IN THE UNITED STATES DISTRICT COURT

FOR THE DISTRICT OF DELAWARE

MOTOROLA, INC.,	)	
	)	
Plaintiff,	)	C. A. No. ________
	)	
v.	)	
	)	**JURY TRIAL DEMANDED**
RESEARCH IN MOTION LIMITED AND RESEARCH IN MOTION CORPORATION,	)	
	)	
	)	
	)	
Defendants.	)	
	)	

COMPLAINT FOR DECLARATORY RELIEF

Plaintiff, Motorola, Inc. ("Motorola"), for its complaint against defendants, Research in Motion Limited ("RIM Ltd.") and Research in Motion Corporation ("RIM Corp.," collectively "Defendants"), hereby demands a jury trial and avers as follows:

JURISDICTION AND VENUE

1. This is an action for patent infringement arising under the patent laws of the United States, 35 U.S.C. §§ 101 *et seq.* This Court has subject matter jurisdiction over this action under 28 U.S.C. §§ 1331, 1338(a) and 2201(a).

2. Venue is proper in this Judicial District under 28 U.S.C. §§ 1391(b), (c), (d), and 1400(b).

3. Upon information and belief, this Court has personal jurisdiction over Defendants, because defendant RIM Corp. is a corporation organized under the laws of the State of Delaware, and because Defendants regularly conduct business in this District.

THE PARTIES

4. Plaintiff Motorola is a corporation organized and existing under the laws of the State of Delaware and having a principal place of business at 1303 East Algonquin Road, Schaumburg, Illinois 60196.

5. Upon information and belief, defendant RIM Ltd. is a corporation organized under the laws of Canada and has its principal place of business at 295 Phillip Street, Waterloo, Ontario, Canada N2L 3WB. Upon information and belief, defendant RIM Ltd. directly or indirectly through its subsidiaries and affiliated companies, distributes, markets, sells and/or offers to sell throughout the United States (including in this Judicial District), and/or imports into the United States products, including wireless communication devices, associated equipment and software.

6. Upon information and belief, defendant RIM Corp. is a corporation organized under the laws of the State of Delaware and has its principal place of business at 122 West John Carpenter Parkway, Suite 430, Irving, Texas 75039. Upon information and belief, defendant RIM Corp. directly or indirectly through its subsidiaries and affiliated companies, distributes, markets, sells and/or offers to sell throughout the United States (including in this Judicial District), and/or imports into the United States products, including wireless communication devices, associated equipment and software.

BACKGROUND

7. In March 2003, Motorola and Defendants entered into a cross-license agreement, whereby the parties agreed to license to each other certain United States Patents relating to cellular telephone technology.

8. In anticipation of that license agreement's December 31, 2007 expiration, the parties engaged in negotiations directed toward an amicable solution.

9. In an effort to avoid paying royalties as part of future license agreements, Defendants asserted that Motorola requires a license to practice several patents in Defendant's portfolio, including patents Defendants recently acquired from Multimedia Patent Trust.

10. Specifically, upon information and belief, Defendants have stated to Motorola that Defendants are the owner by assignment of the following United States Patents ("patents in suit"):

a. United States Patent No. 5,664,055, entitled "CS-ACELP SPEECH COMPRESSION SYSTEM WITH ADAPTIVE PITCH PREDICTION FILTER GAIN BASED ON A MEASURE OF PERIODICITY" ("the '055 patent"), which was issued on September 2, 1997, to Peter Kroon;

b. United States Patent No. 5,699,485, entitled "PITCH DELAY MODIFICATION DURING FRAME ERASURES" ("the '485 patent"), which was issued on December 16, 1997, to Yair Shoham;

c. United States Patent No. 6,611,254 B1, entitled "HAND-HELD ELECTRONIC DEVICE WITH A KEYBOARD OPTIMIZED FOR USE WITH THE THUMBS" ("the '254 patent"), which was issued on August 26, 2003, to Jason T. Griffin, John A. Holmes, Mihal Lazaridis, Herb A. Little, and Harry R. Major;

d. United States Patent No. 6,611,255 B2, entitled "HAND-HELD ELECTRONIC DEVICE WITH A KEYBOARD OPTIMIZED FOR USE WITH THE THUMBS" ("the '255 patent"), which was issued on August 26, 2003, to Jason T. Griffin, John A. Holmes, Mihal Lazaridis, Herb A. Little, and Harry R. Major; and

e. United States Patent No. 6,919,879 B2, entitled "HAND-HELD ELECTRONIC DEVICE WITH A KEYBOARD OPTIMIZED FOR USE WITH THE

THUMBS" ("the '879 patent"), which was issued on July 19, 2005, to Jason T. Griffin, David M. Walters, John A. Holmes, and Mihal Lazaridis.

11. True and correct copies of the patents in suit are attached hereto as Exhibits A–E, respectively.

12. A controversy has arisen as to whether Motorola requires a license to continue engaging in its sale of various mobile communication devices as well as for past sales of such devices. Defendants have represented to Motorola during the negotiations—including as recently as February 8, 2008—that Motorola is on notice of the patents in suit, that Motorola requires a license in order to continue selling various mobile communication devices, and that Defendants are entitled to damages for Motorola's past alleged infringement. Because Motorola is licensed under some of Defendants' patents and because no valid claim of the patents in suit is infringed, Motorola asserts that no additional license is necessary.

CLAIM ONE
(The '055 Patent)

13. Upon information and belief, Motorola is licensed to practice the '055 patent.

14. Upon information and belief, Motorola does not infringe and has not infringed any valid claim of the '055 patent.

15. Upon information and belief, any claim by Defendants against Motorola for past damages for alleged infringement of the '055 patent is barred by the equitable doctrine of laches.

CLAIM TWO
(The '485 Patent)

16. Upon information and belief, Motorola is licensed to practice the '485 patent.

17. Upon information and belief, Motorola does not infringe and has not infringed any valid claim of the '485 patent.

18. Upon information and belief, any claim by Defendants against Motorola for past damages for alleged infringement of the '485 patent is barred by the equitable doctrine of laches.

CLAIM THREE
(The '254 Patent)

19. Upon information and belief, Motorola does not infringe and has not infringed any valid claim of the '254 patent.

CLAIM FOUR
(The '255 Patent)

20. Upon information and belief, Motorola does not infringe and has not infringed any valid claim of the '255 patent.

CLAIM FIVE
(The '879 Patent)

21. Upon information and belief, Motorola does not infringe and has not infringed any valid claim of the '879 patent.

PRAYER FOR RELIEF

WHEREFORE, Motorola prays that the Court enter a judgment against Defendants:

a. Declaring that Motorola has not infringed any valid claim of any of the patents in suit;

b. Declaring that Defendants are not entitled to recover damages for any act of past infringement of any of the patents in suit;

c. Finding this is an exceptional case under 35 U.S.C. § 285;

d. Awarding to Motorola its costs and attorney fees; and

e. Awarding to Motorola such other and further relief as this Court deems proper and just.

DEMAND FOR TRIAL BY JURY

Pursuant to Rule 38(b) of the Federal Rules of Civil Procedure and the Seventh Amendment to the U.S. Constitution, Motorola hereby demands a trial by jury of all claims and all issues triable as of right by jury in this action.

Respectfully submitted,

Josy W. Ingersoll (No. 1088)
Elena C. Norman (No. 4780)
Monté T. Squire (No. 4764)
YOUNG CONAWAY STARGATT &
TAYLOR, LLP
The Brandywine Building
1000 West Street, 17th Floor
Wilmington, DE 19801
Telephone: (302) 571-6600
enorman@ycst.com
Attorneys for Plaintiff Motorola, Inc.

Of Counsel:

Jesse J. Jenner
ROPES & GRAY LLP
1211 Avenue of the Americas
New York, New York 10036

Norman H. Beamer
ROPES & GRAY LLP
525 University Avenue
Palo Alto, CA 94301

Nicole M. Jantzi
ROPES & GRAY LLP
700 12th Street, NW, Suite 900
Washington, DC 20005

Dated: February 16, 2008

EXHIBIT A

US005664055A

United States Patent [19]

Kroon

[11] **Patent Number: 5,664,055**

[45] **Date of Patent: Sep. 2, 1997**

[54] **CS-ACELP SPEECH COMPRESSION SYSTEM WITH ADAPTIVE PITCH PREDICTION FILTER GAIN BASED ON A MEASURE OF PERIODICITY**

[75] Inventor: **Peter Kroon**, Green Grook, N.J.

[73] Assignee: **Lucent Technologies Inc.**, Murray Hill, N.J.

[21] Appl. No.: **482,715**

[22] Filed: **Jun. 7, 1995**

[51] **Int. Cl.**^{6} **G10L 9/14**

[52] **U.S. Cl.** **704/223**; 704/208; 704/219; 704/220; 704/222

[58] **Field of Search** 395/2.16, 2.17, 395/2.28, 2.29, 2.3, 2.31, 2.32, 2.34, 2.33; 381/38, 40

[56] **References Cited**

PUBLICATIONS

Peter Noll, "Digital Audio Coding for Visual Communications", Proc. IEEE, vol. 83, No. 6, pp. 925–943. Jun. 1995.

Allen Gersho, "Advances in Speech and Audio Compression", Proc. IEEE, vol. 82, No. 6, pp. 900–918. Jun. 1994.

Schroeder et al, "Code-Excited Linear Prediction (CELP): High Quality Speech at Very Low Bit Rates," *Proc. ICASSP*, pp. 937–940, 1985.

Kleijn et al, "An Efficient Stochastically Excited Linear Predictive Coding Algorithm for High Quality Low Bit Rate Transmission of Speech," *Speech Communication*, vol. 7, No. 3, pp. 305–316, 1988.

Kroon et al, "On the Use of Pitch Predictors with High Temporal Resolution," *IEEE Trans Signal Proc.*, vol. 39, No. 3, pp. 733–735, 1991.

Tohkura et al., "Spectral smoothing Technique in PARCOR Speech Analysis-Synthesis," *IEEE Trans on ASSP*, vol. 26, No. 6, pp. 587–596. Dec. 1978.

Itakura, "Line Spectrum Representation of Linear Predictive Coefficients of Speech Signals", *J. Acoust. Soc. Amer.*, vol. 57, Suppl. No. 1, S35, 1975.

Soong et al., "Line Spectrum Pair (LSP) and Speech Data Compression," *Proc ICASSP*, pp. 1.10.1–1.10.4, 1984.

Kabal et al., "The Computation of Line Spectral Frequencies Using Chebyshev Polynomials," *IEEE Trans. on ASSP*, vol. 34, No. 6, pp. 1419–1426, Dec. 1986.

Paliwal et al, "Efficient Vector Quantization of LPC parameters at 24 bits/frame," *IEEE Trans. on Acoust. Speech and Signal Proc.*, Jan. 1993, pp. 3–14.

Laflamme et al., "16 kpbs Wideband Speech Coding Technique Based on Algebraic CELP," *Proc. ICASSP*, pp. 13–16.

Primary Examiner—Allen R. MacDonald
Assistant Examiner—Tālivaldis Ivais Smits
Attorney, Agent, or Firm—Thomas A. Restaino; Kenneth M. Brown

[57] **ABSTRACT**

A speech coding system employing an adaptive codebook model of periodicity is augmented with a pitch-predictive filter (PPF). This PPF has a delay equal to the integer component of the pitch-period and a gain which is adaptive based on a measure of periodicity of the speech signal. In accordance with an embodiment of the present invention, speech processing systems which include a first portion comprising an adaptive codebook and corresponding adaptive codebook amplifier and a second portion comprising a fixed codebook coupled to a pitch filter, are adapted to delay the adaptive codebook gain; determine the pitch filter gain based on the delayed adaptive codebook gain, and amplify samples of a signal in the pitch filter based on said determined pitch filter gain. The adaptive codebook gain is delayed for one subframe. The pitch filter gain equals the delayed. adaptive codebook gain, except when the adaptive codebook gain is either less than 0.2 or greater than 0.8., in which cases the pitch filter gain is set equal to 0.2 or 0.8, respectively.

19 Claims, 5 Drawing Sheets

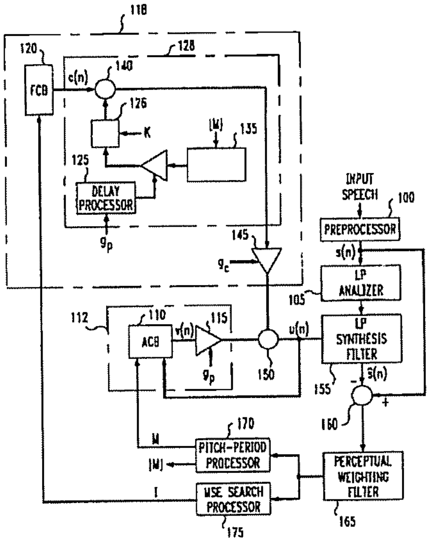


FIG. 1

PRIOR ART

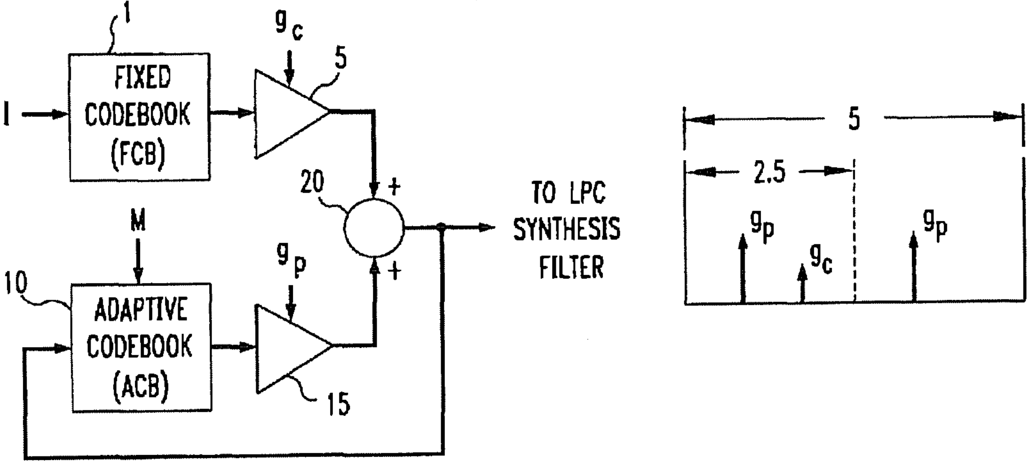


FIG. 2

PRIOR ART

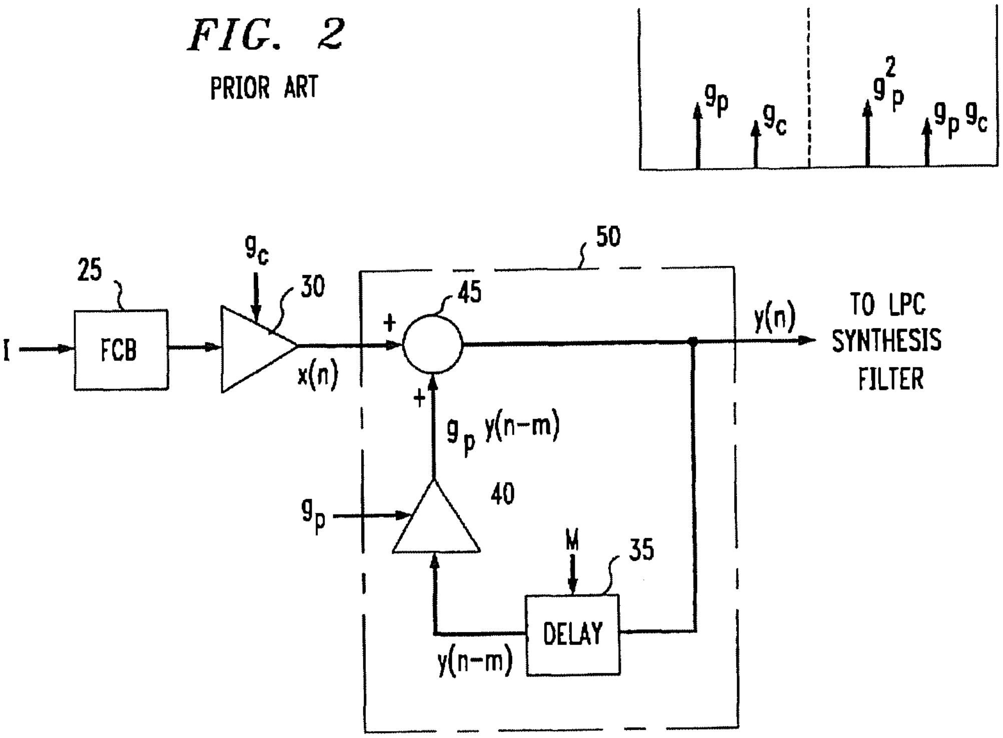


FIG. 3

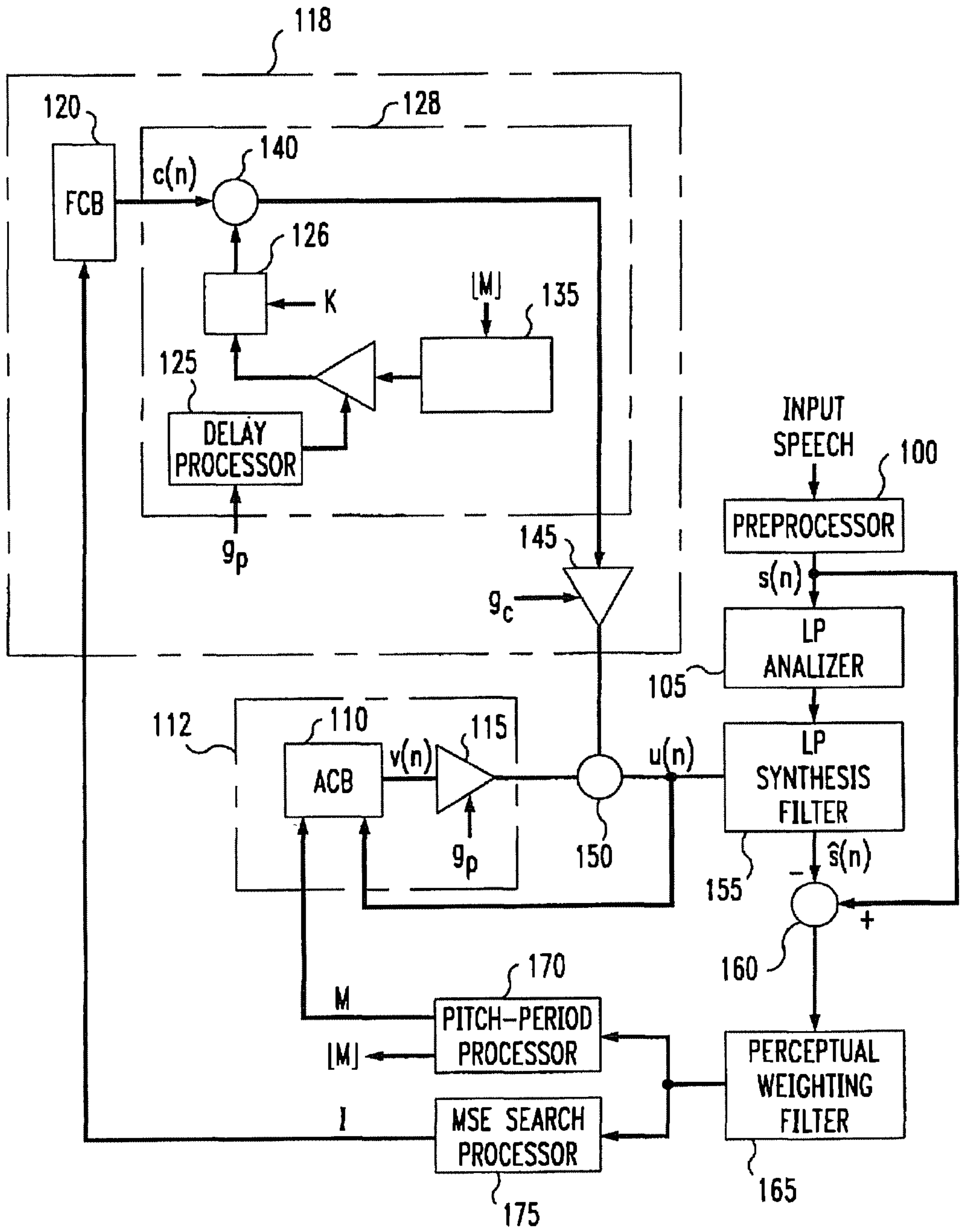

118
120
128
FCB
c(n)
140
126
K
⌊M⌋
135
125
DELAY PROCESSOR
g_p
145
g_c
INPUT SPEECH
100
PREPROCESSOR
s(n)
LP ANALIZER
105
112
110
ACB
v(n)
115
g_p
150
u(n)
LP SYNTHESIS FILTER
155
$\hat{s}(n)$
–
+
160
170
M
PITCH-PERIOD PROCESSOR
⌊M⌋
I
MSE SEARCH PROCESSOR
175
PERCEPTUAL WEIGHTING FILTER
165

FIG. 4

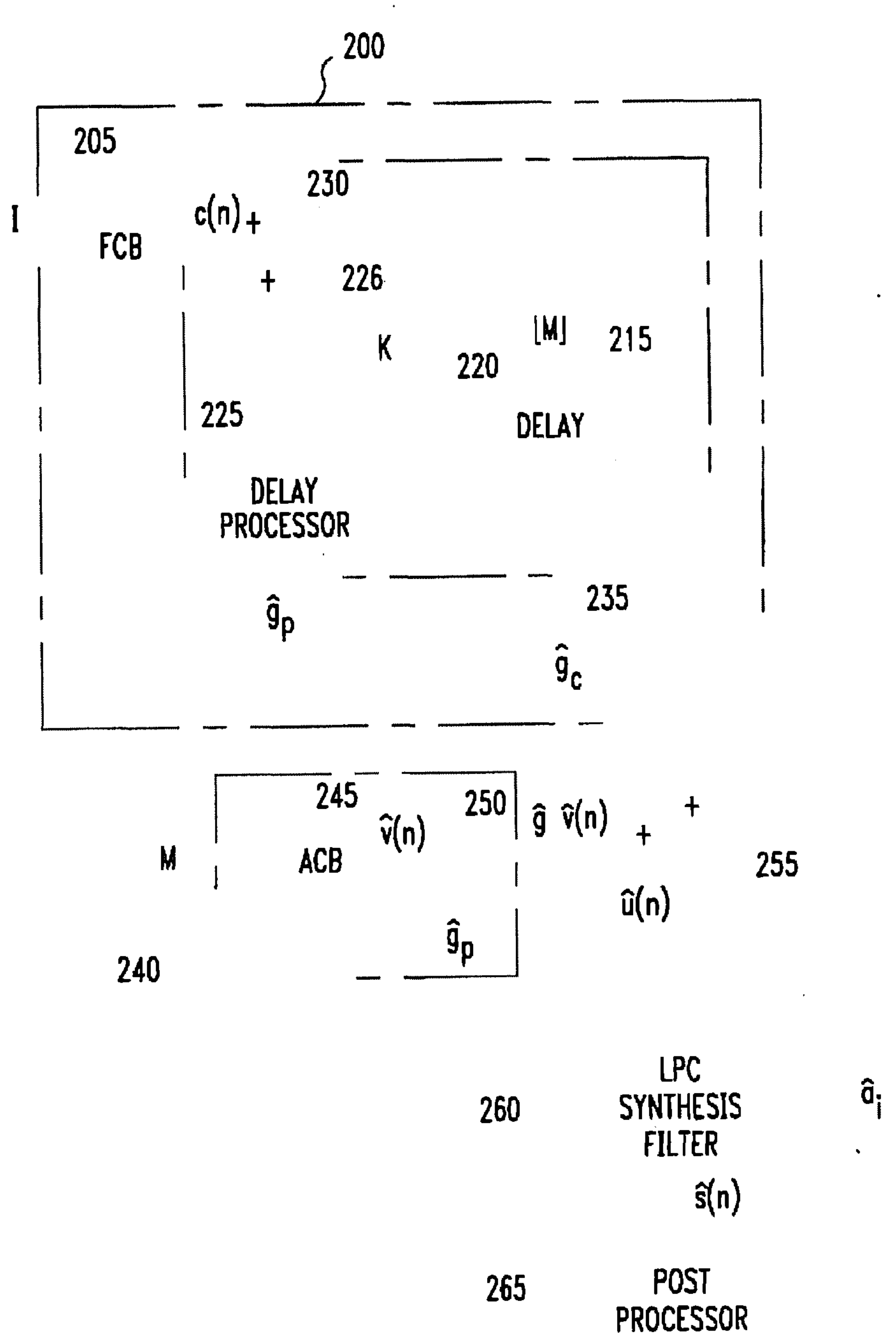

200
205
I
FCB
$c(n)$
+
230
+
226
K
220
[M]
215
DELAY
225
DELAY
PROCESSOR
$\hat{g}_p$
235
$\hat{g}_c$
M
245
ACB
$\hat{v}(n)$
250
$\hat{g}$
$\hat{v}(n)$
+
+
255
$\hat{u}(n)$
$\hat{g}_p$
240
260
LPC
SYNTHESIS
FILTER
$\hat{a}_i$
$\hat{s}(n)$
265
POST
PROCESSOR

FIG. 5

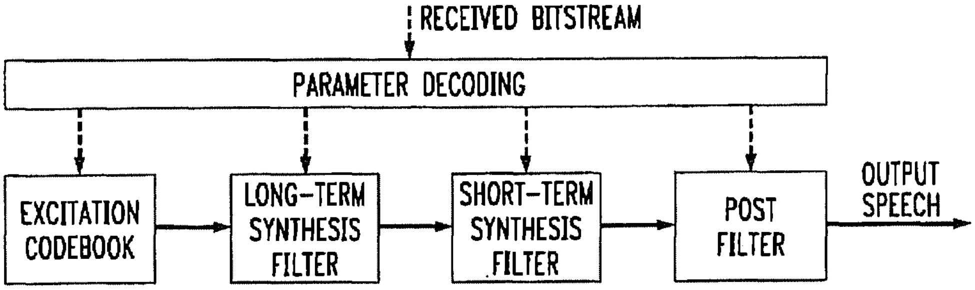

RECEIVED BITSTREAM
PARAMETER DECODING
EXCITATION CODEBOOK
LONG-TERM SYNTHESIS FILTER
SHORT-TERM SYNTHESIS FILTER
POST FILTER
OUTPUT SPEECH

FIG. 6

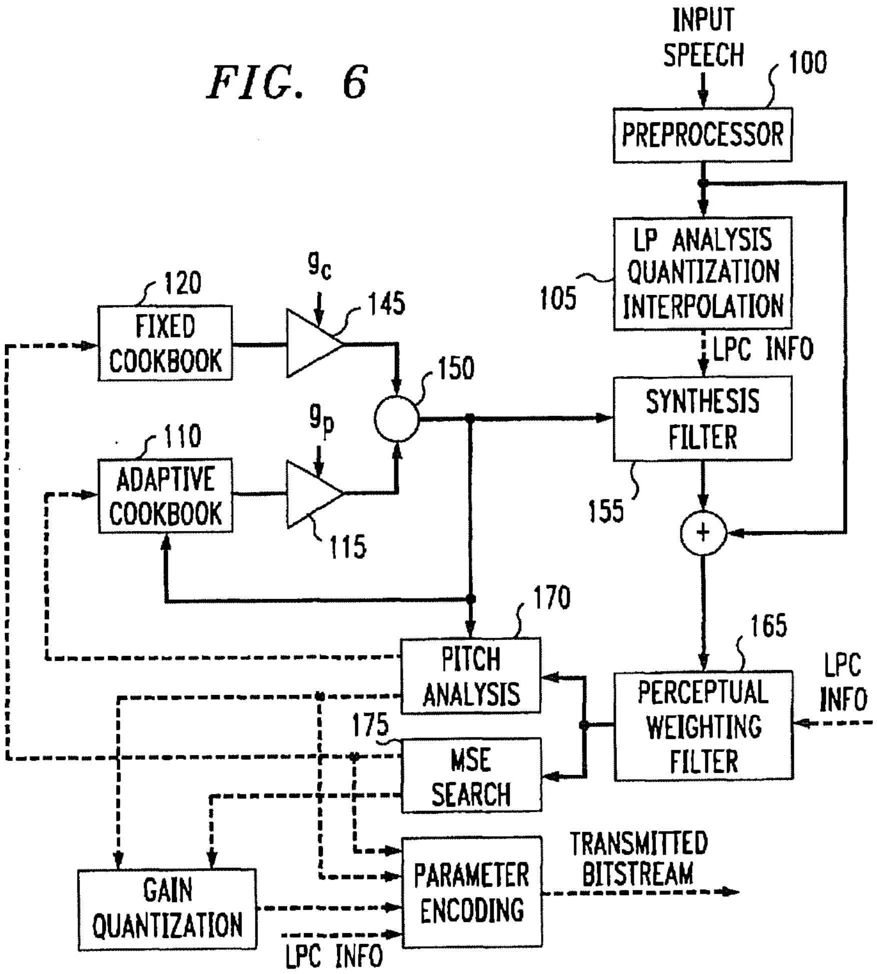

INPUT SPEECH
100
PREPROCESSOR
105
LP ANALYSIS QUANTIZATION INTERPOLATION
LPC INFO
120
FIXED COOKBOOK
g_c
145
150
g_p
110
ADAPTIVE COOKBOOK
115
SYNTHESIS FILTER
155
+
170
PITCH ANALYSIS
165
PERCEPTUAL WEIGHTING FILTER
LPC INFO
175
MSE SEARCH
GAIN QUANTIZATION
PARAMETER ENCODING
TRANSMITTED BITSTREAM
LPC INFO

FIG. 7

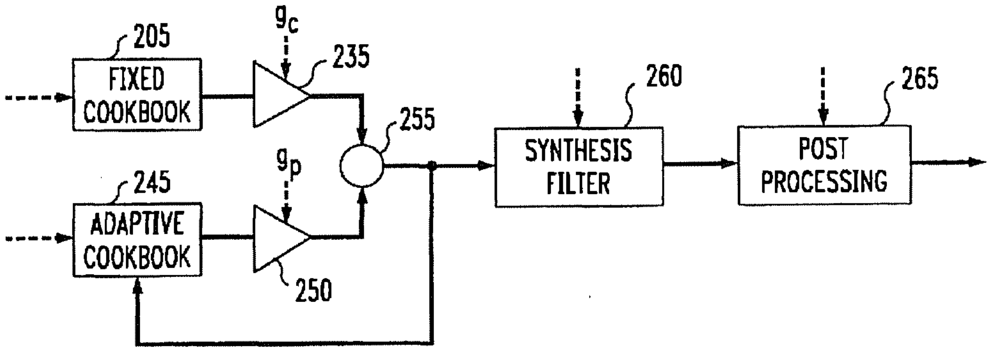
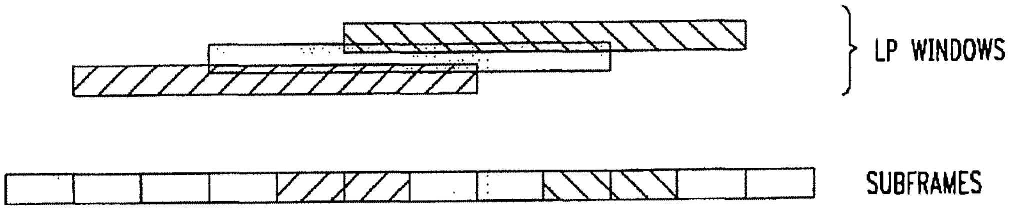

205
FIXED COOKBOOK
g_c
235
255
245
ADAPTIVE COOKBOOK
g_p
250
260
SYNTHESIS FILTER
265
POST PROCESSING

FIG. 8


LP WINDOWS
SUBFRAMES

CS-ACELP SPEECH COMPRESSION SYSTEM WITH ADAPTIVE PITCH PREDICTION FILTER GAIN BASED ON A MEASURE OF PERIODICITY

CROSS-REFERENCE TO RELATED APPLICATION

This application is related to Application Ser. No. 08/485,420, entitled "Codebook Gain Attenuation During Frame Erasure," filed on even date herewith, which is incorporated by reference as if set forth fully herein.

FIELD OF THE INVENTION

The present invention relates generally to adaptive codebook-based speech compression systems, and more particularly to such systems operating to compress speech having a pitch-period less than or equal to adaptive codebook vector (subframe) length.

BACKGROUND OF THE INVENTION

Many speech compression systems employ a subsystem to model the periodicity of a speech signal. Two such periodicity models in wide use in speech compression (or coding) systems are the pitch prediction filter (PPF) and the adaptive codebook (ACB).

The ACB is fundamentally a memory which stores samples of past speech signals, or derivatives thereof such as speech residual or excitation signals (hereafter speech signals). Periodicity is introduced (or modeled) by copying samples from the past (as stored in the memory) speech signal into the present to "predict" what the present speech signal will look like.

The PPF is a simple IIR filter which is typically of the form

$$y(n)=x(n)+g_p y(n-M) \qquad (1)$$

where n is a sample index, y is the output, x is the input, M is a delay value of the filter, and g_p is a scale factor (or gain). Because the current outpbt of the PPF is dependent on a past output, is introduced the PPF.

Although either the ACB or PPF can be used in speech coding, these periodicity models do not operate identically under all circumstances. For example, while a PPF and an ACB will yield the same results when the pitch-period of voiced speech is greater than or equal to the subframe (or codebook vector) size, this is not the case if the pitch-period is less than the subframe size. This difference is illustrated by FIGS. **1** and **2**, where it is assumed that the pitch-period (or delay) is 2.5 ms, but the subframe size is 5 ms.

FIG. 1 presents a conventional combination of a fixed codebook (FCB) and an ACB as used in a typical CELP speech compression system (this combination is used in both the encoder and decoder of the CELP system). As shown in the Figure, FCB **1** receives an index value, I, which causes the FCB to output a speech signal (excitation) vector of a predetermined duration. This duration is referred to as a subframe (here, 5 ms.). Illustratively, this speech excitation signal will consist of one or more main pulses located in the subframe. For purposes of clarity of presentation, the output vector will be assumed to have a single large pulse of unit magnitude. The output vector is scaled by a gain, g_c, applied by amplifier **5**.

In parallel with the operation of the FCB **1** and gain **5**, ACB **10** generates a speech signal based on previously synthesized speech. In a conventional fashion, the ACB **10** searches its memory of past speech for samples of speech which most closely match the original speech being coded. Such samples are in the neighborhood of one pitch-period (M) in the past from the present sample it is attempting to synthesize. Such past speech samples may not exist if the pitch is fractional; they may have to be synthesized by the ACB from surrounding speech sample values by linear interpolation, as is conventional. The ACB uses a past sample identified (or synthesized) in this way as the current sample. For clarity of explanation, the balance of this discussion will assume that the pitch-period is an integral multiple of the sample period and that past samples are identified by M for copying into the present subframe. The ACB outputs individual samples in this manner for the entire subframe (5 ms.). All samples produced by the ACB are scaled by a gain, g_p, applied by amplifier **15**.

For current samples in the second half of the subframe, the "past" samples used as the "current" samples are those samples in the first half of the subframe. This is because the subframe is 5 ms in duration, but the pitch-period, M,—the time period used to identify past samples to use as current samples—is 2.5 ms. Therefore, if the current sample to be synthesized is at the 4 ms point in the subframe, the past sample of speech is at the 4 ms −2.5 ms or 1.5 ms point in the same subframe.

The output signals of the FCB and ACB amplifiers **5**, **15** are summed at summing circuit **20** to yield an excitation signal for a conventional linear predictive (LPC) synthesis filter (not shown). A stylized representation of one subframe of this excitation signal produced by circuit **20** is also shown in FIG. **1**. Assuming pulses of unit magnitudes before scaling, the system of codebooks yields several pulses in the 5 ms subframe. A first pulse of height g_p, a second pulse of height g_c, and a third pulse of height g_p. The third pulse is simply a copy of the first pulse created by the ACB. Note that there is no copy of the second pulse in the second half of the subframe since the ACB memory does not include the second pulse (and the fixed codebook has but one pulse per subframe).

FIG. 2 presents a periodicity model comprising a FCB **25** in series with a PPF **50**. The PPF **50** comprises a summing circuit **45**, a delay memory **35**, and an amplifier **40**. As with the system discussed above, an index, I, applied to the FCB **25** causes the FCB to output an excitation vector corresponding to the index. This vector has one major pulse. The vector is scaled by amplifier **30** which applies gain g_c. The scaled vector is then applied to the PPF **50**. PPF **50** operates according to equation (1) above. A stylized representation of one subframe of PPF **50** output signal is also presented in FIG. **2**. The first pulse of the PPF output subframe is the result of a delay, M, applied to a major pulse (assumed to have unit amplitude) from the previous subframe (not shown). The next pulse in the subframe is a pulse contained in the FCB output vector scaled by amplifier **30**. Then, due to the delay **35** of 2.5 ms, these two pulses are repeated 2.5 ms later, respectively, scaled by amplifier **40**.

There are major differences between the output signals of the ACB and PPF implementations of the periodicity model. They manifest themselves in the later half of the synthesized subframes depicted in FIGS. **1** and **2**. First, the amplitudes of the third pulses are different—g_p as compared with g_p^2. Second, there is no fourth pulse in output of the ACB model. Regarding this missing pulse, when the pitch-period is less than the frame size, the combination of an ACB and a FCB will not introduce a second fixed codebook contribution in the subframe. This is unlike the operation of a pitch prediction filter in series with a fixed codebook.

SUMMARY OF THE INVENTION

For those speech coding systems which employ an ACB model of periodicity, it has been proposed that a PPF be used at the output of the FCB. This PPF has a delay equal to the integer component of the pitch-period and a fixed gain of 0.8. The PPF does accomplish the insertion of the missing FCB pulse in the subframe, but with a gain value which is speculative. The reason the gain is speculative is that joint quantization of the ACB and FCB gains prevents the determination of an ACB gain for the current subframe until both ACB and FCB vectors have been determined.

The inventor of the present invention has recognized that the fixed-gain aspect of the pitch loop added to an ACB based synthesizer results in synthesized speech which is too periodic at times, resulting in an unnatural "buzzyness" of the synthesized speech.

The present invention solves a shortcoming of the proposed use of a PPF at the output of the FCB in systems which employ an ACB. The present invention provides a gain for the PPF which is not fixed, but adaptive based on a measure of periodicity of the speech signal. The adaptive PPF gain enhances PPF performance in that the gain is small when the speech signal is not very periodic and large when the speech signal is highly periodic. This adaptability avoids the "buzzyness" problem.

In accordance with an embodiment of the present invention, speech processing systems which include a first portion comprising an adaptive codebook and corresponding adaptive codebook amplifier and a second portion comprising a fixed codebook coupled to a pitch filter, are adapted to delay the adaptive codebook gain; determine the pitch filter gain based on the delayed adaptive codebook gain, and amplify samples of a signal in the pitch filter based on said determined pitch filter gain. The adaptive codebook gain is delayed for one subframe. The delayed gain is used since the quantized gain for the adaptive codebook is not available until the fixed codebook gain is determined. The pitch filter gain equals the delayed adaptive codebook gain, except when the adaptive codebook gain is either less than 0.2 or greater than 0.8, in which cases the pitch filter gain is set equal to 0.2 or 0.8, respectively. The limits are there to limit perceptually undesirable effects due to errors in estimating how periodic the excitation signal actually is.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 presents a conventional combination of FCB and ACB systems as used in a typical CELP speech compression system, as well as a stylized representation of one subframe of an excitation signal generated by the combination.

FIG. 2 presents a periodicity model comprising a FCB and a PPF, as well as a stylized representation of one subframe of PPF output signal.

FIG. 3 presents an illustrative embodiment of a speech encoder in accordance with the present invention.

FIG. 4 presents an illustrative embodiment of a decoder in accordance with the present invention.

FIG. 5 presents a block diagram of a conceptual G.729 CELP synthesis model.

FIG. 6 presents the signal flow at the G.729 CS-ACELP encoder.

DETAILED DESCRIPTION

I.1 Introduction to the Illustrative Embodiments

For clarity of explanation, the illustrative embodiments of the present invention is presented as comprising individual functional blocks (including functional blocks labeled as "processors"). The functions these blocks represent may be provided through the use of either shared or dedicated hardware, including, but not limited to, hardware capable of executing software. For example, the functions of processors presented in FIG. 3 and 4 may be provided by a single shared processor. (Use of the term "processor" should not be construed to refer exclusively to hardware capable of executing software.)

Illustrative embodiments may comprise digital signal processor (DSP) hardware, such as the AT&T DSP16 or DSP32C, read-only memory (ROM) for storing software performing the operations discussed below, and random access memory (RAM) for storing DSP results. Very large scale integration (VLSI) hardware embodiments, as well as custom VLSI circuitry in combination with a general purpose DSP circuit, may also be provided.

The embodiments described below are suitable for use in many speech compression systems such as, for example, that described in a preliminary Draft Recommendation G.729 to the ITU Standards Body (G.729 Draft), which is provided in Section II. This speech compression system operates at 8 kbit/s is based on Code-Excited Linear-Predictive (CELP) coding. See G.729 Draft Subsection II.2. This draft recommendation includes a complete description of the speech coding system, as well as the use of the present invention therein. See generally, for example, FIG. 6 and the discussion at Subsection II.2.1 of the G.729 Draft. With respect to the an embodiment of present invention, see the discussion at Subsections II.3.8 and II.4.1.2 of the G.729 Draft.

I.2: The Illustrative Embodiments

FIGS. 3 and 4 present illustrative embodiments of the present invention as used in the encoder and decoder of the G.729 Draft. FIG. 3 is a modified version of FIG. 6, which shows the signal flow at the G.729 CS-ACELP encoder. FIG. 3 has been augmented to show the detail of the illustrative encoder embodiment. FIG. 4 is similar to FIG. 7, which shows signal flow at the G.729 CS-ACELP decoder. FIG. 4 is augmented to show the details of the illustrative decoder embodiment. In the discussion which follows, reference will be made to Subsections of the G.729 Draft where appropriate. A general description of the encoder of the G.279 Draft is presented at Subsection II.2.1, while a general description of the decoder is presented at Subsection II.2.2.

A. The Encoder

In accordance with the embodiment, an input speech signal (16 bit PCM at 8 kHz sampling rate) is provided to a preprocessor **100**. Preprocessor **100** high-pass filters the speech signal to remove undesirable low frequency components and scales the speech signal to avoid processing overflow. See Subsection II.3.1. The preprocessed speech signal, s(n), is then provided to linear prediction analyzer **105**. See Subsection II.3.2. Linear prediction (LP) coefficients, $\bar{a}_i$, are provided to LP synthesis filter **155** which receives an excitation signal, u(n), formed of the combined output of FCB and ACB portions of the encoder. The excitation signal is chosen by using an analysis-by-synthesis search procedure in which the error between the original and synthesized speech is minimized according to a perceptually weighted distortion measure by perceptual weighting filter **165**. See Subsection II.3.3

Regarding the ACB portion **112** of the embodiment, a signal representing the perceptually weighted distortion (error) is used by pitch period processor **170** to determine an open-loop pitch-period (delay) used by the adaptive codebook system **110**. The encoder uses the determined open-loop pitch-period as the basis of a closed-loop pitch search. ACB **110** computes an adaptive codebook vector, V(n), by interpolating the past excitation at a selected fractional pitch. See Subsection II.3.4–II.3.7. The adaptive codebook gain amplifier **115** applies a scale factor $\hat{g}_p$ to the output of the ACB system **110**. See Subsection II.3.9.2.

Regarding the FCB portion **118** of the embodiment, an index generated by the mean squared error (MSE) search processor **175** is received by the FCB system **120** and a codebook vector, c(n), is generated in response. See Subsection II.3.8. This codebook vector is provided to the PPF system **128** operating in accordance with the present invention (see discussion below). The output of the PPF system **128** is scaled by FCB amplifier **145** which applies a scale factor $\hat{g}_c$. Scale factor $\hat{g}_c$ is determined in accordance with Subsection II.3.9.

The vectors output from the ACB and FCB portions **112**, **118** of the encoder are summed at summer **150** and provided to the LP synthesis filter as discussed above.

B. The PPF System

As mentioned above, the PPF system addresses the shortcoming of the ACB system exhibited when the pitch-period of the speech being synthesized is less than the size of the subframe and the fixed PPF gain is too large for speech which is not very periodic.

PPF system **128** includes a switch **126** which controls whether the PPF **128** contributes to the excitation signal. If the delay, M, is less than the size of the subframe, L, than the switch **126** is closed and PPF **128** contributes to the excitation. If M≧L, switch **126** is open and the PPF **128** does not contribute to the excitation. A switch control signal K is set when M<L. Note that use of switch **126** is merely illustrative. Many alternative designs are possible, including, for example, a switch which is used to by-pass PPF **128** entirely when M≧L.

The delay used by the PPF system is the integer portion of the pitch-period, M, as computed by pitch-period processor **170**. The memory of delay processor **135** is cleared prior to PPF **128** operation on each subframe. The gain applied by the PPF system is provided by delay processor **125**. Processor **125** receives the ACB gain, $\hat{g}_p$, and stores it for one subframe (one subframe delay). The stored gain value is then compared with upper and lower limits of 0.8 and 0.2, respectively. Should the stored value of the gain be either greater than the upper limit or less than the lower limit, the gain is set to the respective limit. In other words, the PPF gain is limited to a range of values greater than or equal to 0.2 and less than or equal to 0.8. Within that range, the gain may assume the value of the delayed adaptive codebook gain.

The upper and lower limits are placed on the value of the adaptive PPF gain so that the synthesized signal is neither overperiodic or aperiodic, which are both perceptually undesirable. As such, extremely small or large values of the ACB gain should be avoided.

It will be apparent to those of ordinary skill in the art that ACB gain could be limited to the specified range prior to storage for a subframe. As such, the processor stores a signal reflecting the ACB gain, whether pre- or post-limited to the specified range. Also, the exact value of the upper and lower limits are a matter of choice which may be varied to achieve desired results in any specific realization of the present invention.

C. The Decoder

The encoder described above (and in the referenced subsections of the G.729 Draft provided in Section II of this specification provides a frame of data representing compressed speech every 10 ms. The frame comprises 80 bits and is detailed in Tables 1 and 9 of the G.729 Draft. Each 80-bit frame of compressed speech is sent over a communication channel to a decoder which synthesizes a speech (representing two subframes) signals based on the frame produced by the encoder. The channel over which the frames are communicated (not shown) may be of any type (such as conventional telephone networks, cellular or wireless networks, ATM networks, etc.) and/or may comprise a storage medium (such as magnetic storage, semiconductor RAM or ROM, optical storage such as CD-ROM, etc.).

An illustrative decoder in accordance with the present invention is presented in FIG. **4**. The decoder is much like the encoder of FIG. **3** in that it includes both an adaptive codebook portion **240** and a fixed codebook portion **200**. The decoder decodes transmitted parameters (see Subsection II.4.1) and performs synthesis to obtain reconstructed speech.

The FCB portion includes a FCB **205** responsive to a FCB index, I, communicated to the decoder from the encoder. The FCB **205** generates a vector, c(n), of length equal to a subframe. See Subsection II.4.1.3. This vector is applied to the PPF **210** of the decoder. The PPF **210** operates as described above (based on a value of ACB gain, $\hat{g}_p$, delayed in delay processor **225** and ACB pitch-period, M, both received from the encoder via the channel) to yield a vector for application to the FCB gain amplifier **235**. The amplifier, which applies a gain, $\hat{g}_c$, from the channel, generates a scaled version of the vector produced by the PPF **210**. See Subsection II.4.1.4. The output signal of the amplifier **235** is supplied to summer **255** which generates an excitation signal, u(n).

Also provided to the summer **255** is the output signal generated by the ACB portion **240** of the decoder. The ACB portion **240** comprises the ACB **245** which generates an adaptive codebook contribution, v(n), of length equal to a subframe based on past excitation signals and the ACB pitch-period, M, received from encoder via the channel. See Subsection II.4.1.2. This vector is scaled by amplifier **250** based on gain factor, $\hat{g}_p$ received over the channel. This scaled vector is the output of ACB portion **240**.

The excitation signal, u(n), produced by summer **255** is applied to an LPC synthesis filter **260** which synthesizes a speech signal based on LPC coefficients, $\hat{a}_i$, received over the channel. See Subsection II.4.1.6.

Finally, the output of the LPC synthesis filter **260** is supplied to a post processor **265** which performs adaptive postfiltering (see Subsections II.4.2.1–II.4.2.4), high-pass filtering (see Subsection II.4.2.5), and up-scaling (see Subsection II.4.2.5).

I.3 Discussion

Although a number of specific embodiments of this invention have been shown and described herein, it is to be understood that these embodiments are merely illustrative of the many possible specific arrangements which can be devised in application of the principles of the invention. Numerous and varied other arrangements can be devised in accordance with these principles by those of ordinary skill in the art without departing from the spirit and scope of the invention.

For example, should scalar gain quantization be employed, the gain of the PPF may be adapted based on the current, rather than the previous, ACB gain. Also, the values of the limits on the PPF gain (0.2, 0.8) are merely illustrative. Other limits, such as 0.1 and 0.7 could suffice.

In addition, although the illustrative embodiment of present invention refers to codebook "amplifiers," it will be understood by those of ordinary skill in the art that this term encompasses the scaling of digital signals. Moreover, such scaling may be accomplished with scale factors (or gains) which are less than or equal to one (including negative values), as well as greater than one.

The following Appendix to the Detailed Description contains the G.729 Draft described above. This document, at the time of the filing of the present application, is intended to be submitted to a standards body of The International Telecommunications Union (ITU), and provides a more complete description of an illustrative 8 kbit/s speech coding system which employs, inter alia, the principles of the present invention.

APPENDIX TO THE DETAILED DESCRIPTION

SECTION—Draft Recommenation G.729

Coding of Speech at 8kbit/s Using

Conjugate-Structure-Algebraic-Code-Excited

Linear-Predictive (CS-ACELP) Coding

Jun. 7, 1995—Version 4.0

Study Group 15 Contribution—Q.12/15—Submitted to the International Telecommunication Union—Telecommunications Standardization Sector. Until approved by the ITU, neither the C code nor the test vectors contained herein will be available from the ITU. To obtain the C source code, contact Mr. Gerhard Schroeder (Rapporteur SG15/Q.12) at the Deutsche Telekom AG, Postfach 10003, 64276 Darmstadt, Germany; telephone +49 6151 83 3973; facsimile +49 6151 837828; E-mail; gerhard.schroeder@fz13.fz.dbp.de

II.1 INTRODUCTION

This Recommendation contains the description of an algorithm for the coding of speech signals at 8 kbit/s using Conjugate-Structure-Algebraic-Code-Excited Linear-Predictive (CS-ACELP) coding.

This coder is designed to operate with a digital signal obtained by first performing telephone bandwidth filtering (ITU Rec. G.710) of the analog input signal, then sampling it at 8000 Hz, followed by conversion to 16 bit linear PCM for the input to the encoder. The output of the decoder should be converted back to an analog signal by similar means. Other input/output characteristics, such as those specified by ITU Rec. G.711 for 64 kbit/s PCM data, should be converted to 16 bit linear PCM before encoding, or from 16 bit linear PCM to the appropriate format after decoding. The bitstream from the encoder to the decoder is defined within this standard.

This Recommendation is organized as follows: Subsection II.2 gives a general outline of the SC-ACELP algorithm. In Subsections II.3 and II.4, the CS-ACELP encoder and decoder principles are discussed, respectively. Subsection II.5 describes the software that defines this coder in 16 bit fixed point arithmetic.

II.2 General Description of the Coder

The CS-ACELP coder is based on the code-excited linear-predictive (CELP) coding model. The coder operates on speech frames of 10 ms corresponding to 80 samples at a sampling rate of 8000 samples/sec. For every 10 msec frame, the speech signal is analyzed to extract the parameters of the CELP model (LP filter coefficients, adaptive and fixed codebook indices and gains). These parameters are encoded and transmitted. The bit allocation of the coder parameters is shown in Table 1. At the decoder, these parameters are used to retrieve the excitation and synthesis filter parameters. The speech is reconstructed by filtering this excitation through the LP synthesis filter, as is shown in FIG. 5. The short-term synthesis filter is based on a 10th order linear prediction (LP) filter. The long-term, or pitch synthesis filter is implemented using the so-called adaptive codebook approach for delays less than the subframe length. After computing the reconstructed speech, it is further enhanced by a postfilter.

TABLE 1

Bit allocation of the 8 kbit/s CS-ACELP algorithm (10 msec frame).

Parameter	Codeword	Subframe 1	Subframe 2	Total per frame
LSP	L0, L1, L2, L3			18
Adaptive codebook delay	P1, P2	8	5	13
Delay parity	P0	1		1
Fixed codebook index	C1, C2	13	13	26
Fixed codebook sign	S1, S2	4	4	8
Codebook gains (stage 1)	GA1, GA2	3	3	6
Codebook gains (stage 2)	GB1, GB2	4	4	8
				—
Total				80

II.2.1 Encoder

The signal flow at the encoder is shown in FIG. 6. The input signal is high-pass filtered and scaled in the pre-processing block. The pre-processed signal serves as the input signal for all subsequent analysis. LP analysis is done once per 10 ms frame to compute the LP filter coefficients. These coefficients are converted to line spectrum pairs (LSP) and quantized using predictive two-stage vector quantization (VQ) with 18 bits. The excitation sequence is chosen by using an analysis-by-synthesis search procedure in which the error between the original and synthesized speech is minimized according to a perceptually weighted distortion measure. This is done by filtering the error signal with a perceptual weighting filter, whose coefficients are derived from the unquantized LP filter. The amount of perceptual weighting is made adaptive to improve the performance for input signals with a flat frequency-response.

The excitation parameters (fixed and adaptive codebook parameters) are determined per subframe of 5 ms (40 samples) each. The quantized and unquantized LP filter coefficients are used for the second subframe, while in the first subframe interpolated LP filter coefficients are used (both quantized and unquantized). An open-loop pitch delay is estimated once per 10 ms frame based on the perceptually weighted speech signal. Then the following operations are repeated for each subframe. The target signal x(n) is computed by filtering the LP residual through the weighted synthesis filter W(z)/Â(z). The initial states of these filters are updated by filtering the error between LP residual and excitation. This is equivalent to the common approach of subtracting the zero-input response of the weighted synthesis filter from the weighted speech signal. The impulse response, h(n), of the weighted synthesis filter is computed. Closed-loop pitch analysis is then done (to find the adaptive codebook delay and gain), using the target x(n) and impulse response h(n), by searching around the value of the open-loop pitch delay. A fractional pitch delay with ⅓ resolution is used. The pitch delay is encoded with 8 bits in the first subframe and differentially encoded with 5 bits in the second subframe. The target signal x(n) is updated by removing the adaptive codebook contribution (filtered adaptive codevector), and this new target, $x_2(n)$, is used in the fixed algebraic codebook search (to find the optimum excitation). An algebraic codebook with 17 bits is used for the fixed codebook excitation. The gains of the adaptive and fixed codebook are vector quantized with 7 bits, (with MA prediction applied to the fixed codebook gain). Finally, the filter memories are updated using the determined excitation signal.

2.2 Decoder

The signal flow at the decoder is shown in FIG. 7. First, the parameters indices are extracted from the received bitstream. These indices are decoded to obtain the coder parameters corresponding to a 10 ms speech frame. These parameters are the LSP coefficients, the 2 fractional pitch delays, the 2 fixed codebook vectors, and the 2 sets of adaptive and fixed codebook gains. The LSP coefficients are interpolated and converted to LP filter coefficients for each subframe. Then, for each 40-sample subframe the following steps are done:

the excitation is constructed by adding the adaptive and fixed codebook vectors scaled by their respective gains,

the speech is reconstructed by filtering the excitation through the LP synthesis filter,

the reconstructed speech signal is passed through a post-processing stage, which comprises of an adaptive postfilter based on the long-term and short-term synthesis filters, followed by a high-pass filter and scaling operation.

II.2.3 Delay

This coder encodes speech and other audio signals with 10 ms frames. In addition, there is a look-ahead of 5 ms, resulting in a total algorithmic delay of 15 ms. All additional delays in a practical implementation of this coder are due to:

processing time needed for encoding and decoding operations,

transmission time on the communication link,

multiplexing delay when combining audio data with other data.

II.2.4 Speech Coder Description

The description of the speech coding algorithm of this Recommendation is made in terms of bit-exact, fixed-point mathematical operations. The ANSI C code indicated in Subsection II.5, which constitutes an integral part of this Recommendation, reflects this bit-exact, fixed-point descriptive approach. The mathematical descriptions of the encoder (Subsection II.3), and decoder (Subsection II.4), can be implemented in several other fashions, possibly leading to a codec implementation not complying with this Recommendation. Therefore, the algorithm description of the C code of Subsection II.5 shall take precedence over the mathematical descriptions of Subsection II.3 and II.4 whenever discrepancies are found. A non-exhaustive set of test sequences which can be used in conjunction with the C code are available from the ITU.

2.5 Notational Conventions

Throughout this document it is tried to maintain the following notational conventions.

Codebooks are denoted by caligraphic characters (e.g. C).

Time signals are denoted by the symbol and the sample time index between parenthesis (e.g. s(n)). The symbol n is used as sample instant index.

Superscript time indices (e.g $g^{(m)}$) refer to that variable corresponding to subframe m.

Superscripts identify a particular element in a coefficient array.

A $\hat{0}$ identifies a quantized version of a parameter.

Range notations are done using square brackets, where the boundaries are included (e.g. [0.6, 0.9]).

log denotes a logarithm with base 10.

Table 2 lists the most relevant symbols used throughout this document. A glossary of the most relevant signals is given in Table 3. Table 4 summarizes relevant variables and their dimension. Constant parameters are listed in Table 5. The acronyms used in this Recommendation are summarized in Table 6.

TABLE 2

Glossary of symbols.

Name	Reference	Description
1/A(z)	Eq. (2)	LP synthesis filter
$H_{h1}(z)$	Eq. (1)	input high-pass filter
$H_p(z)$	Eq. (77)	pitch postfilter
$H_f(z)$	Eq. (83)	short-term postfilter
$H_t(z)$	Eq. (85)	tilt-compensation filter
$H_{h2}(z)$	Eq. (90)	output high-pass filter
P(z)	Eq. (46)	pitch filter
W(z)	Eq. (27)	weighting filter

TABLE 3

Glossary of signals.

Name	Description
h(n)	impulse response of weighting and synthesis filters
r(k)	auto-correlation sequence
r'(k)	modified auto-correlation sequence
R(k)	correlation sequence
sw(n)	weighted speech signal
s(n)	speech signal
s'(n)	windowed speech signal
sf(n)	postfiltered output
sf'(n)	gain-scaled postfiltered output
$\hat{s}(n)$	reconstructed speech signal
r(n)	residual signal
x(n)	target signal
$x_2(n)$	second target signal
v(n)	adaptive codebook contribution
c(n)	fixed codebook contribution
y(n)	v(n) * h(n)
z(n)	c(n) * h(n)
u(n)	excitation to LP synthesis filter
d(n)	correlation between target signal and h(n)
ew(n)	error signal

TABLE 4

Glossary of variables.

Name	Size	Description
g_p	1	adaptive codebook gain
g_c	1	fixed codebook gain
g_0	1	modified gain for pitch postfilter
g_{pit}	1	pitch gain for pitch postfilter
g_f	1	gain term short-term postfilter
g_t	1	gain term tilt postfilter
T_{op}	1	open-loop pitch delay
a_i	10	LP coefficients
k_i	10	reflection coefficients
o_i	2	LAR coefficients
w_i	10	LSF normalized frequencies
q_i	10	LSP coefficients
r(k)	11	correlation coefficients
w_i	10	LSP weighting coefficients
l_i	10	LSP quantizer output

TABLE 5

Glossary of constants.

Name	Value	Description
f_s	8000	sampling frequency
f_0	60	bandwidth expansion
γ_1	0.94/0.98	weight factor perceptual weighting filter
γ_2	0.60/[0.4–0.7]	weight factor perceptual weighting filter
γ_n	0.55	weight factor post filter
γ_d	0.70	weight factor post filter
γ_p	0.50	weight factor pitch post filter
γ_t	0.90/0.2	weight factor tilt post filter
C	Table 7	fixed (algebraic) codebook
L0	Section 3.2.4	moving average predictor codebook
L1	Section 3.2.4	First stage LSP codebook
L2	Section 3.2.4	Second stage LSP codebook (low part)
L3	Section 3.2.4	Second stage LSP codebook (high part)
GA	Section 3.9	First stage gain codebook
GB	Section 3.9	Second stage gain codebook
w_{lag}	Eq. (6)	correlation lag window
w_{lp}	Eq. (3)	LPC analysis window

TABLE 6

Glossary of acronyms.

Acronym	Description
CELP	code-excited linear-prediction
MA	moving average
MSB	most significant bit
LP	linear prediction
LSP	line spectral pair
LSF	line spectral frequency
VQ	vector quantization

II.3.0 Functional Description of the Encoder

In this section we describe the different functions of the encoder represented in the blocks of FIG. 5.

II.3.1 Pre-Processing

As stated in Subsection II.2, the input to the speech encoder is assumed to be a 16 bit PCM signal. Two pre-processing functions are applied before the encoding process: 1) signal scaling, and 2) high-pass filtering.

The scaling consists of dividing the input by a factor 2 to reduce the possibility of overflows in the fixed-point implementation. The high-pass filter serves as a precaution against undesired low-frequency components. A second order pole/zero filter with a cutoff frequency of 140 Hz is used. Both the scaling and high-pass filtering are combined by dividing the coefficients at the numerator of this filter by 2. The resulting filter is given by

$$h_{h1}(z)=\frac{0.46363718-0.92724705z^{-1}+0.46363718z^{-2}}{1-1.9059465z^{-1}+0.9114024z^{-2}}. \quad (1)$$

The input signal filtered through $H_{h1}(z)$ is referred to as s(n), and will be used. in all subsequent coder operations.

II.3.2 Linear Prediction Analysis and Quantization

The short-term analysis and synthesis filters are based on 10th order linear prediction (LP) filters. The LP synthesis filter is defined as

$$\frac{1}{\hat{A}(z)}=\frac{1}{1+\sum_{i=1}^{10}\hat{a}_i z^{-i}}, \quad (2)$$

where $\hat{a}_i$, i=1, . . . , 10, are the (quantized) linear prediction (LP) coefficients. Short-term prediction, or linear prediction analysis is performed once per speech frame using the autocorrelation approach with a 30 ms asymmetric window. Every 80 samples (10 ms), the autocorrelation coefficients of windowed speech are computed and converted to the LP coefficients using the Levinson algorithm. Then the LP coefficients are transformed to the LSP domain for quantization and interpolation purposes. The interpolated quantized and unquantized filters are converted back to the LP filter coefficients (to construct the synthesis and weighting filters at each subframe).

II.3.2.1 Windowing and Autocorrelation Computation

The LP analysis window consists of two parts: the first part is half a Hamming window and the second part is a quarter of a cosine function cycle. The window is given by:

$$w_{lp}(n)=\begin{cases}0.54-0.46\cos\left(\frac{2\pi n}{399}\right), & n=0,\ldots,199,\\ \cos\left(\frac{2\pi(n-200)}{159}\right), & n=200,\ldots,239.\end{cases} \quad (3)$$

There is a 5 ms lookahead in the LP analysis which means that 40 samples are needed from the future speech frame. This translates into an extra delay of 5 ms at the encoder stage. The LP analysis window applies to 120 samples from past speech frames, 80 samples from the present speech frame, and 40 samples from the future frame. The windowing in LP analysis is illustrated in FIG. 8.

The autocorrelation coefficients of the windowed speech

$$s'(n)=w_{lp}(n)s(n),\ n=0,\ \ldots,\ 239, \quad (4)$$

are computed by

$$r(k)=\sum_{n=k}^{239}s'(n)s'(n-k),\quad k=0,\ldots,10, \quad (5)$$

To avoid arithmetic problems for low-level input signals the value of r(0) has a lower boundary of r(0)=1.0. A 60 Hz bandwidth expansion is applied, by multiplying the autocorrelation coefficients with

$$w_{lag}(k)=\exp\left[-\frac{1}{2}\left(\frac{2\pi f_0 k}{f_s}\right)^2\right], \quad k=1, \ldots, 10, \tag{6}$$

where f_0=60 Hz is the bandwidth expansion and f_s=8000 Hz is the sampling frequency. Further, r(0) is multiplied by the white noise correction factor 1.0001, which is equivalent to adding a noise floor at −40 dB.

II.3.2.2 Levinson-Durbin Algorithm

The modified autocorrelation coefficients

$$r'(0)=1.001r(0)$$

$$r'(k)=w_{lag}(k)r(k), k=1, \ldots, 10 \tag{7}$$

are used to obtain the LP filter coefficients a_i, i=1, . . . , 10, by solving the set of equations

$$\sum_{i=1}^{10} a_i r'(|i-k|)=-r'(k), \quad k=1, \ldots, 10. \tag{8}$$

The set of equations in (8) is solved using the Levinson-Durbin algorithm. This algorithm uses the following recursion:

```
$E(0) = r'(0)$
for $i = 1$ to 10
    $\alpha_0^{(i-1)} = 1$
    $k_i = -\left[\sum_{j=0}^{i-1} a_j^{(i-1)} r'(i-j)\right] / E(i-1)$
    $a_i^{(i)} = k_i$
    for $j = 1$ to $i-1$
        $a_j^{(i)} = a_j^{(i-1)} + k_i a_{i-j}^{(i-1)}$
    end
    $E(i) = (1-k_i^2)E(i-1)$,
    if $E(i) < 0$ then $E(i) = 0.01$
end
```

The final solution is given as $a_j=a_j^{(10)}$, j=1, . . . , 10.

II.3.2.3 LP to LSP Conversion

The LP filter coefficients a_i, i =1, . . . , 10 are converted to the line spectral pair (LSP) representation for quantization and interpolation purposes. For a 10th order LP filter, the LSP coefficients are defined as the roots of the sum and difference polynomials

$$F'_1(z)=A(z)+z^{-11}A(z^{-1}), \tag{9}$$

and

$$F'_2(z)=A(z)-z^{-11}A(z^{-1}), \tag{10}$$

respectively. The polynomial $F'_1(z)$ is symmetric, and $F'_2(z)$ is antisymmetric. It can be proven that all roots of these polynomials are on the unit circle and they alternate each other. $F'_1(z)$ has a root z=−1(w=π) and $F'_2(z)$ has a root z=1 (w=0). To eliminate these two roots, we define the new polynomials

$$F_1(z)=F'_1(z)/(1+z^{-1}), \tag{11}$$

and

$$F_2(z)=F'_2(z)/(1-z^{-1}). \tag{12}$$

Each polynomial has 5 conjugate roots on the unit circle ($e^{\pm jw_i}$), therefore, the polynomials can be written as

$$F_1(z)=\prod_{i=1,3,\ldots,9}(1-2q_i z^{-1}+z^{-2}) \tag{13}$$

-continued

and

$$F_2(z)=\prod_{i=2,4,\ldots,10}(1-2q_i z^{-1}+z^{-2}), \tag{14}$$

where $q_i=\cos(w_i)$ with w_i being the line spectral frequencies (LSF) and they satisfy the ordering property $0<w_1<w_2<\ldots<w_{10}<\pi$. We refer to q_i as the LSP coefficients in the cosine domain.

Since both polynomials $F_1(z)$ and $F_2(z)$ are symmetric only the first 5 coefficients of each polynomial need to be computed. The coefficients of these polynomials are found by the recursive relations

$$f_1(i+1)=a_{i+1}+a_{10-i}-f_1(i), i=0, \ldots, 4,$$

$$f_2(i+1)=a_{i+1}-a_{10-i}+f_2(i), i=0, \ldots, 4, \tag{15}$$

where $f_1(0)=f_2(0)=1.0$. The LSP coefficients are found by evaluating the polynomials $F_1(z)$ and $F_2(z)$ at 60 points equally spaced between 0 and π and checking for sign changes. A sign change signifies the existence of a root and the sign change interval is then divided 4 times to better track the root. The Chebyshev polynomials are used to evaluate $F_1(z)$ and $F_2(z)$. In this method the roots are found directly in the cosine domain $\{q_i\}$. The polynomials $F_1(z)$ or $F_2(z)$, evaluated at $z=e^{jw}$, can be written as

$$F(w)=2e^{-j5w}C(x), \tag{16}$$

with

$$C(x)=T_5(x)+f(1)T_4(x)+f(2)T_3(x)+f(3)t_2(x)+f(4)T_1(x)+f(5)/2, \tag{17}$$

where $T_m(x)=\cos(mw)$ is the ruth order Chebyshev polynomial, and f(i), i=1, . . . , 5, are the coefficients of either $F_1(z)$ or $F_2(z)$, computed using the equations in (15). The polynomial C(x) is evaluated at a certain value of x=cos(w) using the recursive relation:

```
for $k = 4$ downto 1
Kroon 4
    $b_k = 2xb_{k+1} - b_{k+2} + f(5-k)$
end
$C(x) = xb_1 - b_2 + f(5)/2$
```

with initial values b_5=1 and b_6=0.

II.3.2.4 Quantization of the LSP Coefficients

The LP filter coefficients are quantized using the LSP representation in the frequency domain; that is

$$w_i=\arccos(q_i), i=1, \ldots, 10, \tag{18}$$

where w_i are the line spectral frequencies (LSF) in the normalized frequency domain [0, π]. A switched 4th order MA prediction is used to predict the current set of LSF coefficients. The difference between the computed and predicted set of coefficients is quantized using a two-stage vector quantizer. The first stage is a 10-dimensional VQ using codebook L1 with 128 entries (7 bits). The second stage is a 10 bit VQ which has been implemented as a split VQ using two 5-dimensional codebooks, L2 and L3 containing 32 entries (5 bits) each.

To explain the quantization process, it is convenient to first describe the decoding process. Each coefficient is obtained from the sum of 2 codebooks:

$$l_i=\begin{cases} L1_i(L1)+L2_i(L2) & i=1,\ldots,5, \\ L1_i(L1)+L3_{(i-5)}(L3) & i=6,\ldots,10, \end{cases} \tag{19}$$

where L1, L2, and L3 are the codebook indices. To avoid sharp resonances in the quantized LP synthesis filters, the coefficients l_i are arranged such that adjacent coefficients have a minimum distance of J. The rearrangement routine is shown below:

```
for $i = 2, \ldots 10$
  if$(l_{i-1} > l_i - J)$
     $l_{i-1} = (l_i + l_{i-1} - J)/2$
     $l_i = (l_i + l_{i-1} + J)/2$
  end
end
```

This rearrangement process is executed twice. First with a value of J=0.0001, then with a value of J=0.000095.

After this rearrangement process, the quantized LSF coefficients $\hat{w}_i^{(m)}$ for the current frame n, are obtained from the weighted sum of previous quantizer outputs $l^{(m-k)}$, and the current quantizer output $l^{(m)}$

$$\hat{\omega}_i^{(m)}=\left(1-\sum_{k=1}^{4} m_i^k\right) l_i^{(m)}+\sum_{k=1}^{4} m_i^k l_i^{(m-k)},\ i=1,\ldots,10, \tag{20}$$

where m_i^k are the coefficients of the switched MA predictor. Which MA predictor to use is defined by a separate bit L0. At startup the initial values of $l_i^{(k)}$ are given by $l_i=i\pi/11$ for all k <0.

After computing $\hat{w}_i$, the corresponding filter is checked for stability. This is done as follows:

1. Order the coefficient $\hat{w}_i$ in increasing value,
2. If $\hat{w}_1<0.005$ then $\hat{w}_i=0.005$,
3. If $\hat{w}_{i+1}-\hat{w}_i<0.0001$, then $\hat{w}_{i+1}=\hat{w}_{i+}0.0001$ i=1, . . . ,9,
4. If $\hat{w}_{10}>3.135$ then $\hat{w}_{10}=3.135$.

The procedure for encoding the LSF parameters can be outlined as follows. For each of the two MA predictors the best approximation to the current LSF vector has to be found. The best approximation is defined as the one that minimizes a weighted mean-squared error

$$E_{LPC}=\sum_{i=1}^{10} \omega_i(\omega_i-\omega_i)^2. \tag{21}$$

The weights w_i are made adaptive as a function of the unquantized LSF coefficients.

$$w_1=\begin{cases} 1.0 & \text{if } \omega_2-0.04\pi-1>0, \\ 10\,(\omega_2-0.04\pi-1)^2+1 & \text{otherwise} \end{cases} \tag{22}$$

$$w_i\ 2\leqq i\leqq 9=\begin{cases} 1.0 & \text{if } \omega_{i+1}-\omega_{i-1}-1>0, \\ 10\,(\omega_{i+1}-\omega_{i-1}-1)^2+1 & \text{otherwise} \end{cases}$$

$$w_{10}=\begin{cases} 1.0 & \text{if } -\omega_9+0.92\pi-1>0, \\ 10\,(-\omega_9+0.92\pi-1)^2+1 & \text{otherwise} \end{cases}$$

In addition, the weights w_5 and w_6 are multiplied by 1.2 each.

The vector to be quantized for the current frame is obtained from

$$l_i'=\left[\hat{\omega}_i^{(m)}-\sum_{k=1}^{4} m_i^k l_i^{(m-k)}\right] / \left(1-\sum_{k=1}^{4} m_i^k\right),\ i=1,\ldots,10. \tag{23}$$

The first codebook L1 is searched and the entry L1 that minimizes the (unweighted) mean-squared error is selected. This is followed by a search of the second codebook L2, which defines the lower part of the second stage. For each possible candidate, the partial vector $\hat{w}_i$, i=1, . . . ,5 is reconstructed using Eq. (20), and rearranged to guarantee a minimum distance of 0.0001. The vector with index L2 which after addition to the first stage candidate and rearranging, approximates the lower part of the corresponding target best in the weighted MSE sense is selected. Using the selected first stage vector L1 and the lower part of the second stage (L2), the higher part of the second stage is searched from codebook L3. Again the rearrangement procedure is used to guarantee a minimum distance of 0.0001. The vector L3 that minimizes the overall weighted MSE is selected.

This process is done for each of the two MA predictors defined by L0, and the MA predictor L0 that produces the lowest weighted MSE is selected.

II.3.2.5 Interpolation of the LSP Coefficients

The quantized (and unquantized) LP coefficients are used for the second subframe. For the first subframe, the quantized (and unquantized) LP coefficients are obtained from linear interpolation of the corresponding parameters in the adjacent subframes. The interpolation is done on the LSP coefficients in the q domain. Let $q_i^{(m)}$ be the LSP coefficients at the 2nd subframe of frame m, and $q_i^{(m-1)}$ the LSP coefficients at the 2nd subframe of the past frame (m−1). The (unquantized) interpolated LSP coefficients in each of the 2 subframes are given by

$$\text{Subframe 1:}\quad q1_i=0.5q_i^{(m-1)}+0.5q_i^{(m)},\qquad i=1,\ldots,10, \tag{24}$$

$$\text{Subframe 2:}\quad q2_i=q_i^{(m)}\qquad i=1,\ldots,10.$$

The same interpolation procedure is used for the interpolation of the quantized LSP coefficients by substituting q_i by $\hat{q}_i$ in Eq. (24).

II.3.2.6 LSP to LP Conversion

Once the LSP coefficients are quantized and interpolated, they are converted back to LP coefficients $\{a_i\}$. The conversion to the LP domain is done as follows. The coefficients of $F_1(z)$ and $F_2(z)$ are found by expanding Eqs. (13) and (14) knowing the quantized and interpolated LSP coefficients. The following recurslye relation is used to compute $f_1(i)$, i=1, . . . 5, from q_i

```
for $i = 1$ to 5
  $f_1(i) = -2q_{2i-1}f_1(i-1) + 2f_1(i-2)$
  for $j = i-1$ downto 1
    $f_1(j) = f_1(j) - 2q_{2i-1}f_1(j-1) + f_1(j-2)$
  end
end
```

with initial values $f_1(0)=1$ and $f_1(-1)=0$. The coefficients $f_2(i)$ are computed similarly by replacing q_{2i-1} by q_{2i}.

Once the coefficients $f_1(i)$ and $f_2(i)$ are found. $F_1(z)$ and $F_2(z)$ are multiplied by $1+z^{-1}$ and $1-z^{-1}$ respectively, to obtain $F'_1(z)$ and $F'_2(z)$; that is

$$f'_1(i)=f_1(i)+f_1(i-1),\ i=1,\ldots,5,$$

$$f'_2(i)=f_2(i)-f_2(i-1),\ i=1,\ldots,5. \tag{25}$$

Finally the LP coefficients are found by

$$a_i=\begin{cases} 0.5f_1'(i)+0.5f_2'(i), & i=1,\ldots,5, \\ 0.5f_1'(i-5)-0.5f_2'(i-5), & i=6,\ldots,10. \end{cases} \tag{26}$$

This is directly derived from the relation $A(z)=(F'_1(z)+F'_2(z))/2$; and because $F'_1(z)$ and $F'_2(z)$ are symmetric and antisymmetric polynomials, respectively.

II.3.3 Perceptual Weighting

The perceptual weighting filter is based on the unquantized LP filter coefficients and is given by

$$W(z)=\frac{A(z/\gamma_1)}{A(z/\gamma_2)}=\frac{1+\sum_{i=1}^{10}\gamma_1^i a_i z^{-i}}{1+\sum_{i=1}^{10}\gamma_2^i a_i z^{-i}}, \tag{27}$$

The values of γ_1 and γ_2 determine the frequency response of the filter W(z). By proper adjustment of these variables it is possible to make the weighting more effective. This is accomplished by making γ_1 and γ_2 a function of the spectral shape of the input signal. This adaptation is done per 10 ms frame, but an interpolation procedure for each first subframe is used to smooth this adaptation process. The spectral shape is obtained from a 2nd-order lineax prediction filter, obtained as a by product from the Levinson-Durbin recursion (Subsection II.3.2.2). The reflection coefficients k_i, are converted to Log Area Ratio (LAR) coefficients o_i by

$$o_i=\log\frac{(1.0+k_i)}{(1.0-k_i)}\quad i=1,2. \tag{28}$$

These LAR coefficients are used for the second subframe. The LAR, coefficients for the first subframe are obtained through linear interpolation with the LAR parameters from the previous frame, and are given by:

$$\text{Subframe 1:}\quad o1_i=0.5o_i^{(m-1)}+0.5o_i^{(m)},\qquad i=1,\ldots,2, \tag{29}$$

$$\text{Subframe 2:}\quad o2_i=o_i^{(m)},\qquad i=1,\ldots,2.$$

The spectral envelope is characterized as being either flat (flat=1) or tilted (flat=0). For each subframe this characterization is obtained by applying a threshold function to the LAR coefficients. To avoid rapid changes, a hysteresis is used by taking into account the value of flat in the previous subframe (m−1).

$$\text{flat}^{(m)}=\begin{cases}0 & \text{if } o_1<-1.74 \text{ and } o_2>0.65 \text{ and flat}^{(m-1)}=1,\\ 1 & \text{if } o_1>-1.52 \text{ and } o_2<0.43 \text{ and flat}^{(m-1)}=0,\\ \text{flat}^{(m-1)} & \text{otherwise.}\end{cases} \tag{30}$$

If the interpolated spectrum for a subframe is classified as flat (flat$^{(m)}$=1), the weight factors are set to γ_1=0.94 and γ_2=0.6. If the spectrum is classified as tilted (flat$^{(m)}$=0), the value of γ_1, is set to 0.98, and the value of γ_2 is adapted to the strength of the resonances in the LP synthesis filter, but is bounded between 0.4 and 0.7. If a strong resonance is present, the value of γ_2 is set closer to the upperbound. This adaptation is achieved by a criterion based on the minimum distance between 2 successive LSP coefficients for the current subframe. The minimum distance is given by

$$d_{min}=min[w_{i+1}-w_i]i=1,\ldots,9. \tag{31}$$

The following linear relation is used to compute γ_2:

$$\gamma_2=-6.0{*}d_{min}+1.0,\ and\ 0.4\leqq\gamma_2\leqq 0.7 \tag{32}$$

The weighted speech signal in a subframe is given by

$$sw(n)=s(n)+\sum_{i=1}^{10}a_i\gamma_1^i s(n-i)-\sum_{i=1}^{10}a_i\gamma_2^i sw(n-i),\ n=0,\ldots,39. \tag{33}$$

The weighted speech signal sw(n) is used to find an estimation of the pitch delay in the speech frame.

II.3.4 Open-Loop Pitch Analysis

To reduce the complexity of the search for the best adaptive codebook delay, the search range is limited around a candidate delay T_{op}, obtained from an open-loop pitch analysis.

This open-loop pitch analysis is done once per frame (10 ms). The open-loop pitch estimation uses the weighted speech signal sw(n) of Eq. (33), and is done as follows: In the first step, 3 maxima of the correlation

$$R(k)=\sum_{n=0}^{79}sw(n)sw(n-k) \tag{34}$$

are found in the following three ranges

$$i=1:\ 80,\ldots,143,$$
$$i=2:\ 40,\ldots,79,$$
$$i=3:\ 20,\ldots,39.$$

The retained maxima R(t_i), i=1, . . . ,3, are normalized through

$$R'(t_i)=\frac{R(t_i)}{\sqrt{\sum_n sw^2(n-t_i)}},\ i=1,\ldots,3, \tag{35}$$

The winner among the three normalized correlations is selected by favoring the delays with the values in the lower range. This is done by weighting the normalized correlations correspondiffg to the longer delays. The best open-loop delay T_{op} is determined as follows:

```
$T_{op} = t_1$
$R'(T_{op}) = R'(t_1)$
if $R'(t_2) \geqq 0.85R'(T_{op})$
    $R'(T_{op}) = R'(t_2)$
    $T_{op} = t_2$
end
if $R'(t_3) \geqq 0.85R'(T_{op})$
    $R'(T_{op}) = R'(t_3)$
    $T_{op} = t_3$
end
```

This procedure of dividing the delay range into 3 sections and favoring the lower sections is used to avoid choosing pitch multiples.

II3.5 Computation of the Impulse Response

The impulse response, h(n), of the weighted synthesis filter W(z)/Â(z) is computed for each subframe. This impulse response is needed for the search of adaptive and fixed codebooks. The impulse response h(n) is computed by filtering the vector of coefficients of the filter A(z/γ_1) extended by zeros through the two filters 1/Â(z) and 1/A(z/γ_2).

II3.6 Computation of the Target Signal

The target signal x(n) for the adaptive codebook search is usually computed by subtracting the zero-input response of the weighted synthesis filter W(z)/Â(z)=A(z/γ_1)/[Â(z)A(z/γ_2)] from the weighted speech signal sw(n) of Eq. (33). This is done on a subframe basis.

An equivalent procedure for computing the target signal, which is used in this Recommendation, is the filtering of the LP residual signal r(n) through the combination of synthesis filter 1/Â(z) and the weighting filter A(z/γ_1)/A(z/γ_2). After determining the excitation for the subframe, the initial states of these filters are updated by filtering the difference between the LP residual and excitation. The memory update of these filters is explained in Subsection II.3.10.

The residual signal r(n), which is needed for finding the target vector is also used in the adaptive codebook search to extend the past excitation buffer. This simplifies the adaptive codebook search procedure for delays less than the subframe size of 40 as will be explained in the next section. The LP residual is given by

$$r(n)=s(n)+\sum_{i=1}^{10}\hat{a}_i s(n-i),\ n=0,\ldots,39. \tag{36}$$

II.3.7 Adaptive-Codebook Search

The adaptive-codebook parameters (or pitch parameters) are the delay and gain. In the adaptive codebook approach for implementing the pitch filter, the excitation is repeated for delays less than the subframe length. In the search stage, the excitation is extended by the LP residual to simplify the closed-loop search. The adaptive-codebook search is done every (5 ms) subframe. In the first subframe, a fractional pitch delay T_1 is used with a resolution of ⅓ in the range [19⅓, 84⅔] and integers only in the range [85, 143]. For the second subframe, a delay T_2 with a resolution of ⅓ is always used in the range [(int)T_1−5⅔, (int)T_1+4⅔], where (int)T_1 is the nearest integer to the fractional pitch delay T_1 of the first subframe. This range is adapted for the cases where T_1 straddles the boundaries of the delay range.

For each subframe the optimal delay is determined using closed-loop analysis that minimizes the weighted mean-squared error. In the first subframe the delay T_1 is found be searching a small range (6 samples) of delay values around the open-loop delay T_{op} (see Subsection II.3.4). The search boundaries t_{min} and t_{max} are defined by

```
$t_{min} = T_{op} - 3$
if $t_{min} < 20$ then $t_{min} = 20$
$t_{max} = t_{min} + 6$
if $t_{max} > 143$ then
        $t_{max} = 143$
        $t_{min} = t_{max} - 6$
end
```

For the second subframe, closed-loop pitch analysis is done around the pitch selected in the first subframe to find the optimal delay T_2. The search boundaries are between t_{min}−⅔ and t_{max}+⅔, where t_{min} and t_{max} are derived from T_1 as follows:

```
$t_{min} = (\mathrm{int})T_1 - 5$
if $t_{min} < 20$ then $t_{min} = 20$
$t_{max} = t_{min} + 9$
if $t_{max} > 143$ then
        $t_{max} = 143$
        $t_{min} = t_{max} - 9$
end
```

The closed-loop pitch search minimizes the mean-squared weighted error between the original and synthesized speech. This is achieved by maximizing the term

$$R(k)=\frac{\sum_{n=0}^{39} x(n)y_k(n)}{\sqrt{\sum_{n=0}^{39} y_k(n)y_k(n)}}, \tag{37}$$

where x(n) is the target signal and $y_k(n)$ is the past filtered excitation at delay k (past excitation convolved with h(n)). Note that the search range is limited around a preselected value, which is the open-loop pitch T_{op} for the first subframe, and T_1 for the second subframe.

The convolution $y_k(n)$ is computed for the delay t_{min}, and for the other integer delays in the search range k=t_{min}+1, . . . ,t_{max}, it is updated using the recursive relation

$$y_k(n)=y_{k-1}(n-1)+u(-k)h(n),\ n=39,\ldots,0, \tag{38}$$

where u(n), n=−143, . . . , 39, is the excitation buffer, and $y_{k-1}(-1)=0$. Note that in the search stage, the samples u(n), n=0, . . . , 39 are not known, and they are needed for pitch delays less than 40. To simplify the search, the LP residual is copied to u(n) to make the relation in Eq. (38) valid for all delays.

For the determination of T_2, and T_1 if the optimum integer closed-loop delay is less than 84, the fractions around the optimum integer delay have to be tested. The fractional pitch search is done by interpolating the normalized correlation in Eq. (37) and searching for its maximum. The interpolation is done using a FIR filter b_{12} based on a Hamming windowed sine function with the sinc truncated at ±11 and padded with zeros at ±12 ($b_{12}(12)=0$). The filter has its cut-off frequency (3 dB) at 3600 Hz in the oversampled domain. The interpolated values of R(k) for the fractions −⅔, −⅓, 0, ⅓, and ⅔ are obtained using the interpolation formula

$$R(k)_t=\sum_{i=0}^{3} R(k-i)b_{12}(t+i\cdot 3)+\sum_{i=0}^{3} R(k+1+i)b_{12}(3-t+i\cdot 3), \tag{39}$$
$$t=0,\ 1,\ 2,$$

where t=0, 1, 2 corresponds to the fractions 0, ⅓, and ⅔, respectively. Note that it is necessary to compute correlation terms in Eq. (37) using a range t_{min}−4, t_{max}+4, to allow for the proper interpolation.

II.3.7.1 Generation of the Adaptive Codebook Vector

Once the noninteger pitch delay has been determined, the adaptive codebook vector v(n) is computed by interpolating the past excitation signal u(n) at the given integer delay k and fraction t

$$v(n)=\sum_{i=0}^{9} u(n-k+i)b_{30}(t+i\cdot 3)+ \tag{40}$$
$$\sum_{i=0}^{9} u(n-k+1+i)b_{30}(3-t+i\cdot 3),\ n=0,\ldots,39,\ t=0,1,2.$$

The interpolation filter b_{30} is based on a Hamming windowed sine functions with the sine truncated at ±29 and padded with zeros at ±30 ($b_{30}(30)=0$). The filters has a cut-off frequency (−3 dB) at 3600 Hz in the oversampled domain.

II3.7.2 Codeword Computation for Adaptive Codebook Delays

The pitch delay T_1 is encoded with 8 bits in the first subframe and the relative delay in the second subframe is encoded with 5 bits. A fractional delay T is represented by its integer part (int)T, and a fractional part frac/3, frac=−1, 0,1. The pitch index P1, is now encoded as

$$P1= \tag{41}$$
$$\begin{cases} ((\mathrm{int})T_1-19)*3+\mathrm{frac}-1, & \text{if } T_1=[19,\ldots,85],\ \mathrm{frac}=[-1,0,1] \\ ((\mathrm{int})T_1-85)+197, & \text{if } T_1=[86,\ldots,143],\ \mathrm{frac}=0 \end{cases}$$

The value of the pitch delay T_2 is encoded relative to the value of T_1. Using the same interpretation as before, the fractional delay T_2 represented by its integer part (int)T_2, and a fractional part frac/3, frac=−1,0,1, is encoded as

$$P2=((\mathrm{int})T_2-t_{min})*3+frac+2 \tag{42}$$

where t_{min} is derived from T_1 as before.

To make the coder more robust against random bit errors, a parity bit P0 is computed on the delay index of the first subframe. The parity bit is generated through an XOR operation on the 6 most significant bits of P1. At the decoder this parity bit is recomputed and if the recomputed value does not agree with the transmitted value, an error concealment procedure is applied.

II.3.7.3 Computation of the Adaptive-Codebook Gain

Once the adaptive-codebook delay is determined, the adaptive-codebook gain g_p is computed as

$$g_p = \frac{\sum_{n=0}^{39} x(n)y(n)}{\sum_{n=0}^{39} y(n)y(n)}, \text{ bounded by } 0 \leqq g_p \leqq 1.2, \tag{43}$$

where y(n) is the filtered adaptive codebook vector (zero-state response of W(z)/Â(z) to v(n)). This vector is obtained by convolving v(n) with h(n)

$$y(n) = \sum_{i=0}^{n} v(i)h(n-i)\ n = 0, \ldots, 39. \tag{44}$$

Note that by maximizing the term in Eq. (37) in most cases g_p>0. In case the signal contains only negative correlations, the value of g_p is set to 0.

II.3.8 Fixed Codebook: Structure and Search

The fixed codebook is based on an algebraic codebook structure using an interleaved single-pulse permutation (ISPP) design. In this codebook, each codebook vector contains 4 non-zero pulses. Each pulse can have either the amplitudes +1 or −1, and can assume the positions given in Table 7.

The codebook vector c(n) is constructed by taking a zero vector, and putting the 4 unit pulses at the found locations, multiplied with their corresponding sign.

$$c(n)=s0\delta(n-i0)+s1\delta(n-i1)+s2\delta(n-i2)+s3\delta(n-i3),\ n=0, \ldots, 39. \tag{45}$$

where δ(0) is a unit pulse. A special feature incorporated in the codebook is that the selected codebook vector is filtered through an adaptive pre-filter P(z) which enhances harmonic components to improve the synthesized speech quality. Here the filter

$$P(z)=1/(1-\beta z^{-T}) \tag{46}$$

is used, where T is the integer component of the pitch delay of the current subframe, and β is a pitch gain. The value of β is made adaptive by using the quantized adaptive codebook gain from the previous subframe bounded by 0.2 and 0.8.

TABLE 7

Structure of fixed codebook C.

Pulse	Sign	Positions
i0	s0	0, 5, 10, 15, 20, 25, 30, 35
i1	s1	1, 6, 11, 16, 21, 26, 31, 36
i2	s2	2, 7, 12, 17, 22, 27, 32, 37
i3	s3	3, 8, 13, 18, 23, 28, 33, 38
		4, 9, 14, 19, 24, 29, 34, 39

$$\beta=g_p^{(m-1)},\ 0.2 \leqq \beta \leqq 0.8. \tag{47}$$

This filter enhances the harmonic structure for delays less than the subframe size of 40. This modification is incorporated in the fixed codebook search by modifying the impulse response h(n), according to

$$h(n)=h(n)+\beta h(n-t),\ n=T, \ldots, 39. \tag{48}$$

II.3.8.1 Fixed-Codebook Search Procedure

The fixed codebook is searched by minimizing the mean-squared error between the weighted input speech sw(n) of Eq. (33), and the weighted reconstructed speech. The target signal used in the closed-loop pitch search is updated by subtracting the adaptive codebook contribution. That is

$$x_2(n)=x(n)-g_p y(n),\ n=0, \ldots, 39, \tag{49}$$

where y(n) is the filtered adaptive codebook vector of Eq. (44).

The matrix H is defined as the lower triangular Toepliz convolution matrix with diagonal h(0) and lower diagonals h(1), . . . , h(39). If c_k is the algebraic codevector at index k, then the codebook is searched by maximizing the term

$$\frac{C_k^2}{E_k} = \frac{\left(\sum_{n=0}^{39} d(n)c_k(n)\right)^2}{c_k{}^t\Phi c_k}, \tag{50}$$

where d(n) is the correlation between the target signal x_2(n) and the impulse response h(n), and Φ=H'H is the matrix of correlations of h(n). The signal d(n) and the matrix Φ are computed before the codebook search. The elements of d(n) are computed from

$$d(n) = \sum_{i=n}^{39} x(i)h(i-n),\ n = 0, \ldots, 39, \tag{51}$$

and the elements of the symmetric matrix Φ are computed by

$$\phi(i,j) = \sum_{n=j}^{39} h(n-i)h(n-j),\ (j \geqq i), \tag{52}$$

Note that only the elements actually needed are computed and an efficient storage procedure has been designed to speed up the search procedure.

The algebraic structure of the codebook C allows for a fast search procedure since the codebook vector c_k contains only four nonzero pulses. The correlation in the numerator of Eq. (50) for a given vector c_k is given by

$$C = \sum_{i=0}^{3} a_i d(m_i), \tag{53}$$

where m_i is the position of the ith pulse and a_i is its amplitude. The energy in the denominator of Eq. (50) is given by

$$E = \sum_{i=0}^{3} \phi(m_i, m_i) + 2 \sum_{i=0}^{2} \sum_{j=i+1}^{3} a_i a_j \phi(m_i, m_j). \tag{54}$$

To simplify the search procedure, the pulse amplitudes are predetermined by quantizing the signal d(n). This is done by setting the amplitude of a pulse at a certain position equal to the sign of d(n) at that position. Before the codebook search, the following steps are done. First, the signal d(n) is decomposed into two signals: the absolute signal d'(n)=|d(n)| and the sign signal sign[d(n)]. Second, the matrix Φ is modified by including the sign information; that is,

$$\phi'(i,j)=sign[d(i)]sign[d(j)]\phi(i,j),\ i=0, \ldots, 39,\ j=i, \ldots, 39. \tag{55}$$

To remove the factor 2 in Eq. (54)

$$\phi'(i,i)=0.5\phi(i,i),\ i=0, \ldots, 39. \tag{56}$$

The correlation in Eq. (53) is now given by

$$C=d'(m_0)+d'(m_1)+d'(m_2)+d'(m_3), \tag{57}$$

and the energy in Eq. (54) is given by

$$\begin{aligned} E = {} & \phi'(m_0, m_0) + \\ & \phi'(m_1, m_1) + \phi'(m_0, m_1) + \\ & \phi'(m_2, m_2) + \phi'(m_0, m_2) + \phi'(m_1, m_2) + \\ & \phi'(m_3, m_3) + \phi'(m_0, m_3) + \phi'(m_1, m_3) + \phi'(m_2, m_3). \end{aligned} \tag{58}$$

A focused search approach is used to further simplify the search procedurel. In this approach a precomputed threshold is tested before entering the last loop, and the loop is entered only if this threshold is exceeded. The maximum number of times the loop can be entered is fixed so that a low percentage of the codebook is searched. The threshold is computed based on the correlation C. The maximum absolute correlation and the average correlation due to the contribution of the first three pulses, max_3 and av_3, are found before the codebook search. The threshold is given by

$$thr_3=av_3+K_3(max_3-av_3). \tag{59}$$

The fourth loop is entered only if the absolute correlation (due to three pulses) exceeds thr_3, where $0 \leqq K_3 < 1$. The value of K_3 controls the percentage of codebook search and it is set here to 0.4. Note that this results in a variable search time, and to further control the search the number of times the last loop is entered (for the 2 subframes) cannot exceed a certain maximum, which is set here to 180 (the average worst case per subframe is 90 times).

II.3.8.2 Codeword Computation of the Fixed Codebook

The pulse positions of the pulses i0, i1, and i2, are encoded with 3 bits each, while the position of i3 is encoded with 4 bits. Each pulse amplitude is encoded with 1 bit. This gives a total of 17 bits for the 4 pulses. By defining s=1 if the sign is positive and s=0 is the sign is negative, the sign codeword is obtained from

$$S=s0+2*s1+4*s2+8*s3 \tag{60}$$

and the fixed codebook codeword is obtained from

$$C=(i0/5)+8*(i1/5)+64*(i2/5)+512*(2*(i3/5)+jx) \tag{61}$$

where jx=0 if i3=3,8, . . . , and jx=1 if i3=4,9, . . .

II.3.9 Quantization of the Gains

The adaptive-codebook gain (pitch gain) and the fixed (algebraic) codebook gain are vector quantized using 7 bits. The gain codebook search is done by minimizing the mean-squared weighted error between original and reconstructed speech which is given by

$$E=x^tx+g_p{}^2y^ty+g_c{}^2z^tz-2g_px^ty-2g_cx^tz+2g_pg_cy^tz, \tag{62}$$

where x is the target vector (see Subsection II.3.6), y is the filtered adaptive codebook vector of Eq. (44), and z is the fixed codebook vector convolved with h(n),

$$z(n) = \sum_{i=0}^{n} c(i)h(n-i) \quad n = 0, \ldots, 39. \tag{63}$$

II.3.9.1 Gain Prediction

The fixed codebook gain g_c can be expressed as

$$g_c=\gamma g'_c, \tag{64}$$

where g'_c is a predicted gain based on previous fixed codebook energies, and γ is a correction factor.

The mean energy of the fixed codebook contribution is given by

$$E = 10 \log \left(\frac{1}{40} \sum_{i=0}^{39} c_i^2 \right). \tag{65}$$

After scaling the vector c_i with the fixed codebook gain g_c, the energy of the scaled fixed codebook is given by 20 log g_c+E. Let $E^{(m)}$ be the mean-removed energy (in dB) of the (scaled) fixed codebook contribution at subframe m, given by

$$E^{(m)}=20 \log g_c+E-\bar{E}, \tag{66}$$

where $\bar{E}$=30 dB is the mean energy of the fixed codebook excitation. The g_c can be expressed as a function of $E^{(m)}$, E, and $\bar{E}$ by

$$g_c=10^{(E^{(m)}+\bar{E}-E)/20}. \tag{67}$$

The predicted gain g'_c is found by predicting the log-energy of the current fixed codebook contribution from the log-energy of previous fixed codebook contributions. The 4th order MA prediction is done as follows. The predicted energy is given by

$$\tilde{E}^{(m)} = \sum_{i=1}^{4} b_i\hat{R}^{(m-i)}, \tag{68}$$

where $[b_1\ b_2\ b_3\ b_4]=[0.68\ 0.58\ 0.34\ 0.19]$ are the MA prediction coefficients, and $\hat{R}^{(m)}$ is the quantized version of the prediction error $R^{(m)}$ at subframe m, defined by

$$R^{(m)}=E^{(m)}-\tilde{E}^{(m)}. \tag{69}$$

The predicted gain g'_c is found by replacing $E^{(m)}$ by its predicted value in Eq (67).

$$g'_c=10^{(\tilde{E}^{(m)}+\bar{E}-E)/20}. \tag{70}$$

The correction factor γ is related to the gain-prediction error by

$$R^{(m)}=E^{(m)}-\tilde{E}^{(m)}=20 \log (\gamma). \tag{71}$$

II.3.9.2 Codebook Search for Gain Quantization

The adaptive-codebook gain, g_p, and the factor γ are vector quantized using a 2-stage conjugate structured codebook. The first stage consists of a 3 bit two-dimensional codebook GA, and the second stage consists of a 4 bit two-dimensional codebook GB. The first element in each codebook represents the quantized adaptive codebook gain $\hat{g}_p$, and the second element represents the quantized fixed codebook gain correction factor $\hat{\gamma}$. Given codebook indices m and n for GA and GB, respectively, the quantized adaptive-codebook gain is given by

$$\hat{g}_p=GA_1(m)+GB_1(n) \tag{72}$$

and the quantized fixed-codebook gain by

$$\hat{g}_c=g'_c\hat{\gamma}=g'_c(GA_2(m)+GB_2(n)). \tag{73}$$

This conjugate structure simplifies the codebook search, by applying a pre-selection process. The optimum pitch gain g_p, and fixed-codebook gain, g_c, are derived from Eq. (62), and are used for the pre-selection. The codebook GA contains 8 entries in which the second element (correspOnding to g_c) has in general larger values than the first element (corresponding to g_p). This bias alloyes a pre-selection using the value of g_c. In this pre-selection process, a cluster of 4 vectors whose second element are close to gx_c, where gx_c is derived from g_c and g_p. Similarly, the codebook GB contains

16 entries in which have a bias towards the first element (corresponding to g_p). A cluster of 8 vectors whose first elements are close to g_p are selected. Hence for each codebook the best 50% candidate vectors are selected. This is followed by an exhaustive search over the remaining 4* 8=32 possibilities, such that the combination of the two indices minimizes the weighted mean-squared error of Eq. (62).

II.3.9.3 Codeword Computation for Gain Quantizer

The codewords GA and GB for the gain quantizer are obtained from the indices corresponding to the best choice. To reduce the impact of single bit errors the codebook indices are mapped.

II.3.10 Memory Update

An update of the states of the synthesis and weighting filters is needed to compute the target signal in the next subframe. After the two gains are quantized, the excitation signal, u(n), in the present subframe is found by

$$u(n)=\hat{g}_p v(n)+\hat{g}_c c(n),\ n=0,\ \ldots,39, \qquad (74)$$

where $\hat{g}_p$ and $\hat{g}_c$ are the quantized adaptive and fixed codebook gains, respectively, v(n) the adaptive codebook vector (interpolated past excitation), and c(n) is the fixed codebook vector (algebraic codevector including pitch sharpening). The states of the filters can be updated by filtering the signal r(n)−u(n) (difference between residual and excitation) through the filters $1/\hat{A}(z)$ and $A(z/\gamma_1)/A(z/\gamma_2)$ for the 40 sample subframe and saving the states of the filters. This would require 3 filter operations. A simpler approach, which requires only one filtering is as follows. The local synthesis speech, $\hat{s}(n)$, is computed by filtering the excitation signal through $1/\hat{A}(z)$. The output of the filter due to the input r(n)−u(n) is equivalent to $e(n)=s(n)-\hat{s}(n)$. So the states of the synthesis filter $1/\hat{A}(z)$ are given by e(n), n=30, . . . , 39. Updating the states of the filter $A(z/\gamma_1)/A(z/\gamma_2)$ can be done by filtering the error signal e(n)through this filter to find the perceptually weighted error ew(n). However, the signal ew(n) can be equivalently found by

$$ew(n)=x(n)-\hat{g}_p y(n)+\hat{g}_c z(n). \qquad (75)$$

Since the signals x(n), y(n), and z(n) are available, the states of the weighting filter are updated by computing ew(n) as in Eq. (75) for n=30, . . . , 39. This saves two filter operations.

II.3.11 Encoder and Decoder Initialization

All static encoder variables should be initialized to 0, except the variables listed in table 8. These variables need to be initialized for the decoder as well.

TABLE 8

Description of parameters with nonzero initialization.

Variable	Reference	Initial value
β	Section 3.8	0.8
l_i	Section 3.2.4	iπ/11
q_i	Section 3.2.4	0.9595, . . . ,
$\hat{R}^{(k)}$	Section 3.9.1	−14

II.4.0 Functional Description of the Decoder

The signal now at the decoder was shown in Subsection II.2 (FIG. 7). First the parameters are decoded (LP coefficients, adaptive codebook vector, fixed codebook vector, and gains). These decoded parameters are used to compute the reconstructed speech signal. This process is described in Subsection II.4.1. This reconstructed signal is enhanced by a post-processing operation consisting of a postfilter and a high-pass filter (Subsection II.4.2). Subsection II.4.3 describes the error concealment procedure used when either a parity error has occurred, or when the frame erasure flag has been set.

II.4.1 Parameter Decoding Procedure

The transmitted parameters are listed in Table 9. At startup all static encoder variables should be initialized to 0, except the variables listed in Table 8. The decoding process is done in the following order:

TABLE 9

Description of transmitted parameters indices. The bitstream ordering is reflected by the order in the table. For each parameter the most significant bit (MSB) is transmitted first.

Symbol	Description	Bits
L0	Switched predictor index of LSP quantizer	1
L1	First stage vector of LSP quantizer	7
L2	Second stage lower vector of LSP quantizer	5
L3	Second stage higher vector of LSP quantizer	5
P1	Pitch delay 1st subframe	8
P0	Parity bit for pitch	1
S1	Signs of pulses 1st subframe	4
C1	Fixed codebook 1st subframe	13
GA1	Gain codebook (stage 1) 1st subframe	3
GB1	Gain codebook (stage 2) 1st subframe	4
P2	Pitch delay 2nd subframe	5
S2	Signs of pulses 2nd subframe	4
C2	Fixed codebook 2nd subframe	13
GA2	Gain codebook (stage 1) 2nd subframe	3
GB2	Gain codebook (stage 2) 2nd subframe	4

II.4.1.1 Decoding of LP Filter Parameters

The received indices L0, L1, L2, and L3 of the LSP quantizer are used to reconstruct the quantized LSP coefficients using the procedure described in Subsection II.3.2.4. The interpolation procedure described in Subsection II.3.2.5 is used to obtain 2 interpolated LSP vectors (corresponding to 2 subframes). For each subframe, the interpolated LSP vector is converted to LP filter coefficients a_i, which are used for synthesizing the reconstructed speech in the subframe.

The following steps are repeated for each subframe:

1. decoding of the adaptive codebook vector,
2. decoding of the fixed codebook vector,
3. decoding of the adaptive and fixed codebook gains,
4. computation of the reconstructed speech,

II.4.1.2 Decoding of the Adaptive Codebook Vector

The received adaptive codebook index is used to find the integer and fractional parts of the pitch delay. The integer part $(int)T_1$ and fractional part frac of T_1 are obtained from P1 as follows:

```
if $P1 < 197$
    $(int)T_1 = (P1 + 2)/3 + 19$
    $frac = P1 - (int)T_1*3 + 58$
else
    $(int)T_1 = P1 - 112$
    $frac = 0$
end
```

The integer and fractional part of T_2 are obtained from P2 and t_{min}, where t_{min} is derived from P1 as follows

```
$t_{min} = (int)T_1 - 5$
if $t_{min} < 20$ then $t_{min} = 20$
$t_{max} = t_{min} + 9$
if $t_{max} > 143$ then
    $t_{max} = 143$
    $t_{min} = t_{max} - 9$
```

-continued

```
    end
Now $T_2$ is obtained from
    $(\text{int})T_2 = (P2 + 2)/3 - 1 + t_{min}$
    $\text{frac} = P2 - 2 - ((P2 + 2)/3 - 1)*3$
```

The adaptive codebook vector v(n) is found by interpolating the past excitation u(n) (at the pitch delay) using Eq. (40).

II.4.1.3 Decoding of the Fixed Codebook Vector

The received fixed codebook index C is used to extract the positions of the excitation pulses. The pulse signs are obtained from S. Once the pulse positions and signs are decoded the fixed codebook vector c(n), can be constructed. If the integer part of the pitch delay, T, is less than the subframe size 40, the pitch enhancement procedure is applied which modifies c(n) according to Eq. (48).

II.4.1.4 Decoding of the Adaptive and Fixed Codebook Gains

The received gain codebook index gives the adaptive codebook gain $\hat{g}_p$ and the fixed codebook gain correction factor $\hat{\gamma}$. This procedure is described in detail Subsection II.3.9The estimated fixed codebook gain g'_c is found using Eq. (70). The fixed codebook vector is obtained from the product of the quantized gain correction factor with this predicted gain (Eq. (64)). The adaptive codebook gain is reconstructed using Eq. (72).

II.4.1.5 Computation of the Parity Bit

Before the speech is reconstructed, the parity bit is recomputed from the adaptive codebook delay (Subsection II.3.7.2). If this bit is not identical to the transmitted parity bit P0, it is likely that bit errors occurred during transmission and the error concealment procedure of Subsection II.4.3 is used.

II.4.1.6 Computing the Reconstructed Speech

The excitation u(n) at the input of the synthesis filter (see Eq. (74)) is input to the LP synthesis filter. The reconstructed speech for the subframe is given by

$$\hat{s}(n) = u(n) - \sum_{i=1}^{10} \hat{a}_i\hat{s}(n-i), \quad n = 0, \ldots, 39. \tag{76}$$

where $\hat{a}_i$ are the interpolated LP filter coefficients.

The reconstructed speech $\hat{s}(n)$ is then processed by a post processor which is described in the next section.

II.4.2 Post-Processing

Post-processing consists of three functions: adaptive postfiltering, high-pass filtering, and signal up-scaling. The adaptive postfilter is the cascade of three filters: a pitch postfilter $H_p(z)$, a short-term postfilter $H_f(z)$, and a tilt compensation filter $H_t(z)$, followed by an adaptive gain control procedure. The postfilter is updated every subframe of 5 ms. The postfiltering process is organized as follows. First, the synthesis speech $\hat{s}(n)$ is inverse filtered through $\hat{A}(z/\gamma_n)$ to produce the residual signal $\hat{r}(n)$. The signal $\hat{r}(n)$ is used to compute the pitch delay T and gain g_{pit}. The signal $\hat{r}(n)$ is filtered through the pitch postfilter $H_p(z)$ to produce the signal r'(n) which, in its turn, is filtered by the synthesis filter $1/[g_f\hat{A}(z/\gamma_d)]$. Finally, the signal at the output of the synthesis filter $1/[g_f\hat{A}(z/\gamma_d)]$ is passed to the tilt compensation filter $H_t(z)$ resulting in the postfiltered synthesis speech signal sf(n). Adaptive gain controle is then applied between sf(n) and $\hat{s}(n)$ resulting in the signal sf'(n). The high-pass filtering and scaling operation operate on the postfiltered signal sf'(n).

II.4.2.1 Pitch Postfilter

The pitch, or harmonic, postfilter is given by

$$H_p(z) = \frac{1}{1+g_0}\,(1 + g_0 z^{-T}), \tag{77}$$

where T is the pitch delay and g_0 is a gain factor given by

$$g_0 = \gamma_p g_{pit} \tag{78}$$

where g_{pit} is the pitch gain. Both the pitch delay and gain are determined from the decoder output signal. Note that g_{pit} is bounded by 1, and it is set to zero if the pitch prediction gain is less that 3 dB. The factor γ_p controls the amount of harmonic postfiltering and has the value γ_p=0.5. The pitch delay and gain are computed from the residual signal $\hat{r}(n)$ obtained by filtering the speech $\hat{s}(n)$ through $\hat{A}(z/\gamma_n)$, which is the numerator of the short-term postfilter (see Subsection II.4.2.2)

$$\hat{r}(n) = \hat{s}(n) + \sum_{i=1}^{10} \gamma_n^i\hat{a}_i\hat{s}(n-i). \tag{79}$$

The pitch delay is computed using a two pass procedure. The first pass selects the best integer T_0 in the range $[T_1-1, T_1+1]$, where T_1 is the integer part of the (transmitted) pitch delay in the first subframe. The best integer delay is the one that maximizes the correlation

$$R(k) = \sum_{n=0}^{39} \hat{r}(n)\hat{r}(n-k). \tag{80}$$

The second pass chooses the best fractional delay T with resolution ⅛ around T_0. This is done by finding the delay with the highest normalized correlation.

$$R'(k) = \frac{\sum_{n=0}^{39} \hat{r}(n)\hat{r}_k(n)}{\sqrt{\sum_{n=0}^{39} \hat{r}_k(n)\hat{r}_k(n)}}, \tag{81}$$

where $\hat{r}_k(n)$ is the residual signal at delay k. Once the optimal delay T is found, the corresponding correlation value is compared against a threshold. If R'(T)<0.5 then the harmonic postfilter is disabled by setting g_{pit}=0. Otherwise the value of g_{pit} is computed from:

$$g_{pit} = \frac{\sum_{n=0}^{39} \hat{r}(n)\hat{r}_k(n)}{\sum_{n=0}^{39} \hat{r}_k(n)\hat{r}_k(n)}, \quad \text{bounded by } 0 \leqq g_{pit} \leqq 1.0. \tag{82}$$

The noninteger delayed signal $\hat{r}_k(n)$ is first computed using an interpolation filter of length 33. After the selection of T, $\hat{r}_k(n)$ is recomputed with a longer interpolation filter of length 129. The new signal replaces the previous one only if the longer filter increases the value of R'(T).

II.4.2.2 Short-Term Postfilter

The short-term post filter is given by

$$H_f(z) = \frac{1}{g_f}\,\frac{\hat{A}(z/\gamma_n)}{\hat{A}(z/\gamma_d)} = \frac{1}{g_f}\,\frac{1 + \sum_{i=1}^{10} \gamma_n^i\hat{a}_i z^{-i}}{1 + \sum_{i=1}^{10} \gamma_d^i\hat{a}_i z^{-i}}, \tag{83}$$

where $\hat{A}(z)$ is the received quantized LP inverse filter (LP analysis is not done at the decoder), and the factors γ_n and γ_d control the amount of short-term postfiltering, and are set to γ_n=0.55, and γ_d=0.7. The gain term g_f is calculated on the truncated impulse response, $h_f(n)$, of the filter $\hat{A}(z/\gamma_n)/\hat{A}(z/\gamma_d)$ and given by

$$g_f = \sum_{n=0}^{19} |h_f(n)|. \tag{84}$$

II.4.2.3 Tilt Compensation

Finally, the filter $H_t(z)$ compensates for the tilt in the short-term postfilter $H_f(z)$ and is given by

$$H_t(z) = \frac{1}{g_t}\left(1 + \gamma_t k_1 z^{-1}\right), \tag{85}$$

where $\gamma_t k_1$ is a tilt factor, k_1 being the first reflection coefficient calculated on $h_f(n)$ with

$$k_1 = -\frac{r_h(1)}{r_h(0)}\ ; \quad r_h(i) = \sum_{j=0}^{19-i} h_f(j) h_f(j+i). \tag{86}$$

The gain term $g_t = 1 - |\gamma_t k_1|$ compensates for the decreasing effect of gf in $H_f(z)$. Furthermore, it has been shown that the product filter $H_f(z)H_t(z)$ has generally no gain.

Two values for γ_t are used depending on the sign of k_1. If k_1 is negative, $\gamma_t=0.9$, and if k_1 is positive, $\gamma_t=0.2$.

I.I4.2.4 Adaptive Gain Control

Adaptive gain control is used to compensate for gain differences between the reconstructed speech signal $\hat{s}(n)$ and the postfiltered signal sf(n). The gain scaling factor G for the present subframe is computed by

$$G = \frac{\sum_{n=0}^{39} |\hat{s}(n)|}{\sum_{n=0}^{39} |sf(n)|}. \tag{87}$$

The gain-scaled postfiltered signal sf'(n) is given by

$$sf'(n) = g(n)sf(n),\ n=0, \ldots, 39, \tag{88}$$

where g(n) is updated on a sample-by-sample basis and given by

$$g(n) = 0.85g(n-1) + 0.15G,\ n=0, \ldots, 39. \tag{89}$$

The initial value of $g(-1)=1.0$.

II.4.2.5 High-pass Filtering and Up-Scaling

A high-pass filter at a cutoff frequency of 100 Hz is applied to the reconstructed and postfiltered speech sf'(n). The filter is given by

$$H_{h2}(z) = \frac{0.93980581 - 1.8795834z^{-1} + 0.93980581z^{-2}}{1 - 1.9330735z^{-1} + 0.93589199z^{-2}}. \tag{90}$$

Up-scaling consists of multiplying the high-pass filtered output by a factor 2 to retrieve the input signal level.

II.4.3 Concealment of Frame Erasures and Parity Errors

An error concealment procedure has been incorporated in the decoder to reduce the degradations in the reconstructed speech because of frame erasures or random errors in the bitstream. This error concealment process is functional when either i) the franie of coder parameters (corresponding to a 10 ms frame) has been identified as being erased, or ii) a checksum error occurs on the parity bit for the pitch delay index P1. The latter could occur when the bitstream has been corrupted by random bit errors.

If a parity error occurs on P1, the delay value T_1 is set to the value of the delay of the previous frame. The value of T_2 is derived with the procedure outlined in Subsection II.4.1.2, using this new value of T_1. If consecutive parity errors occur, the previous value of T_1, incremented by 1, is used.

The mechanism for detecting frame erasures is not defined in the Recommendation, and will depend on the application. The concealment strategy has to reconstruct the current frame, based on previously received information. The method used replaces the missing excitation signal with one of similar characteristics, while gradually decaying its energy. This is done by using a voicing classifier based on the long-term prediction gain, which is computed as part of the long-term postfilter analysis. The pitch postfilter (see Subsection II.4.2.1) finds the long-term predictor for which the prediction gain is more than 3 dB. This is done by setting a threshold of 0.5 on the normalized correlation R'(k) (Eq. (81)). For the error concealment process, these frames will be classified as periodic. Otherwise the frame is declared nonperiodic. An erased frame inherits its class from the preceding (reconstructed) speech frame. Note that the voicing classification is continuously updated based on this reconstructed speech signal. Hence, for many consecutive erased frames the classification might change. Typically, this only happens if the original classification was periodic.

The specific steps taken for an erased frame are:

1. repetition of the LP filter parameters,
2. attenuation of adaptive and fixed codebook gains,
3. attenuation of the memory of the gain predictor,
4. generation of the replacement excitation.

II.4.3.1 Repetition of LP Filter Parameters

The LP parameters of the last good frame are used. The states of the LSF predictor contain the values of the received codewords l_i. Since the current codeword is not available it is computed from the repeated LSF parameters $\hat{w}_i$ and the predictor memory from

$$l_i = \left[\hat{\omega}_i^{(m)} - \sum_{k=1}^{4} m_i^k l_i^{(m-k)}\right] / \left(1 - \sum_{k=1}^{4} m_i^k\right),\ i=1, \ldots, 10. \tag{91}$$

II.4.3.2 Attenuation of Adaptive and Fixed Codebook Gains

An attenuated version of the previous fixed codebook gain is used.

$$g_c^{(m)} = 0.98g_c^{(m-1)}. \tag{92}$$

The same is done for the adaptive codebook gain. In addition a clipping operation is used to keep its value below 0.9.

$$g_p^{(m)} = 0.9g_p^{(m-1)} \text{ and } g_p^{(m)} < 0.9. \tag{93}$$

II.4.3.3 Attenuation of the Memory of the Gain Predictor

The gain predictor uses the energy of previously selected codebooks. To allow for a smooth continuation of the coder once good frames are received, the memory of the gain predictor is updated with an attenuated version of the codebook energy. The value of $\hat{R}^{(m)}$ for the current subframe n is set to the averaged quantized gain prediction error, attenuated by 4 dB.

$$\hat{R}^{(m)} = \left(0.25 \sum_{i=1}^{4} \hat{R}^{(m-i)}\right) - 4.0 \text{ and } \hat{R}^{(m)} \geqq -14. \tag{94}$$

II.4.3.4 Generation of the Replacement Excitation

The excitation used depends on the periodicity classification. If the last correctly received frame was classified as periodic, the current frame is considered to be periodic as well. In that case only the adaptive codebook is used, and the fixed codebook contribution is set to zero. The pitch delay is based on the last correctly received pitch delay and is repeated for each successive frame. To avoid excessive periodicity the delay is increased by one for each next subframe but bounded by 143. The adaptive codebook gain is based on an attenuated value according to Eq. (93).

If the last correctly received frame was classified as nonperiodic, the current frame is considered to be nonperiodic as well, and the adaptive codebook contribution is set to zero. The fixed codebook contribution is generated by randomly selecting a codebook index and sign index. The random generator is based on the function

$$seed=seed*31821+13849, \qquad (95)$$

with the initial seed value of 21845. The random codebook index is derived from the 13 least significant bits of the next random number. The random sign is derived from the 4 least significant bits of the next random number. The fixed codebook gain is attenuated according to Eq. (92).

II.5.6 Bit-Exact Description of the CS-ACELP Coder

ANSI C code simulating the CS-ACELP coder in 16 bit fixed-point is available from ITU-T. The following sections summarize the use of this simulation code, and how the software is organized.

II.5.1 Use of the Simulation Software

The C code consists of two main programs coder.c, which simulates the encoder, and decoder.c, which simulates the decoder. The encoder is run as follows:

coder inputfile bstreamfile

The inputfile and outputfile are sampled data files containing 16-bit PCM signals. The bitstream file contains 81 16-bit words, where the first word can be used to indicate frame erasure, and the remaining 80 words contain one bit each.

The decoder takes this bitstream file and produces a postfiltered output file containing a 16-bit PCM signal.

decoder bstreamfile outputfile

II.5.2 Organization of the Simulation Software

In the fixed-point ANSI C simulation, only two types of fixed-point data are used as is shown in Table 10. To facilitate the implementation of the simulation code, loop indices, Boolean values and flags use the type Flag, which would be either 16 bit or 32 bits depending on the target platform.

TABLE 10

Data types used in ANSI C simulation.

Type	Max. value	Min. value	Description
Word16	0 × 7fff	0 × 8000	signed 2's complement 16 bit word
Word32	0 × 7fffffffL	0 × 80000000L	signed 2's complement 32 bit word

All the computations are done using a predefined set of basic operators. The description of these operators is given in Table 11. The tables used by the simulation coder are summarized in Table 12. These main programs use a library of routines that are summarized in Tables 13, 14, and 15.

TABLE 11

Kroon 4

Basic operations used in ANSI C simulation.

Operation	Description
Word16 sature(Word32 L_var1)	Limit to 16 bits
Word16 add(Word16 var1, Word16 var2)	Short addition
Word16 sub(Word16 var1, Word16 var2)	Short subtraction
Word16 abs_s(Word16 var1)	Short abs
Word16 shl(Word16 var1, Word16 var2)	Short shift left
Word16 shr(Word16 var1, Word16 var2)	Short shift right
Word16 mult(Word16 var1, Word16 var2)	Short multiplication
Word32 L_mult(Word16 var1, Word16 var2)	Long multiplication
Word16 negate(Word16 var1)	Short negate
Word16 extract_h(Word32 L_var1)	Extract high
Word16 extract_l(Word32 L_var1)	Extract low
Word16 round(Word32 L_var1)	Round
Word32 L_mac(Word32 L_var3, Word16 var1, Word16 var2)	Mac
Word32 L_msu(Word32 L_var3, Word16 var1, Word16 var2)	Msu
Word32 L_macNs(Word32 L_var3, Word16 var1, Word16 var2)	Mac without sat
Word32 L_msuNs(Word32 L_var3, Word16 var1, Word16 var2)	Msu without sat
Word32 L_add(Word32 L_var1, Word32 L_var2)	Long addition
Word32 L_sub(Word32 L_var1, Word32 L_var2)	Long subtraction
Word32 L_add_c(Word32 L_var1, Word32 L_var2)	Long add with c
Word32 L_sub_c(Word32 L_var1, Word32 L_var2)	Long sub with c
Word32 L_negate(Word32 L_var1)	Long negate
Word16 mult_r(Word16 var1, Word16 var2)	Multiplication with round
Word32 L_shl(Word32 L_var1, Word16 var2)	Long shift left
Word32 L_shr(Word32 L_var1, Word16 var2)	Long shift right
Word16 shr_r(Word16 var1, Word16 var2)	Shift right with round
Word16 mac_r(Word32 L_var3, Word16 var1, Word16 var2)	Mac with rounding
Word16 msu_r(Word32 L_var3, Word16 var1, Word16 var2)	Msu with rounding
Word32 L_deposit_h(Word16 var1)	16 bit var1 -ℓ MSB
Word32 L_deposit_l(Word16 var1)	16 bit var1 -ℓ LSB
Word32 L_shr_r(Word32 L_var1, Word16 var2)	Long shift right with round
Word32 L_abs(Word32 L_var1)	Long abs
Word32 L_sat(Word32 L_var1)	Long saturation
Word16 norm_s(Word16 var1)	Short norm
Word16 div_s(Word16 var1, Word16 var2)	Short division
Word16 norm_l(Word32 L_var1)	Long norm

TABLE 12

Summary of tables.

File	Table name	Size	Description
tab_hup.c	tab_hup_s	28	upsampling filter for postfilter
tab_hup.c	tab_hup_l	112	upsampling filter for postfilter
inter_3.c	inter_3	13	FIR filter for interpolating the correlation
pred_lt3.c	inter_3	31	FIR filter for interpolating past excitation
lspcb.tab	lspcb1	128 × 10	LSP quantizer (first stage)
lspcb.tab	lspcb2	32 × 10	LSP quantizer (second stage)
lspcb.tab	fg	2 × 4 × 10	MA predictors in LSP VQ
lspcb.tab	fg_sum	2 × 10	used in LSP VQ
lspcb.tab	fg_sum_inv	2 × 10	used in LSP VQ
qua_gain.tab	gbk1	8 × 2	codebook GA in gain VQ
qua_gain.tab	gbk2	16 × 2	codebook GB in gain VQ
qua_gain.tab	map1	8	used in gain VQ
qua_gain.tab	imap1	8	used in gain VQ
qua_gain.tab	map2	16	used in gain VQ
qua_gain.tab	ima21	16	used in gain VQ
window.tab	window	240	LP analysis window
lag_wind.tab	lag_h	10	lag window for bandwidth expansion (high part)
lag_wind.tab	lag_l	10	lag window for bandwidth expansion (low part)
grid.tab	grid	61	grid points in LP to LSP conversion
inv_sqrt.tab	table	49	lookup table in inverse square root computation
log2.tab	table	33	lookup table in base 2 logarithm computation
lsp_lsf.tab	table	65	lookup table in LSF to LSP conversion and vice versa
lsp_lsf.tab	slope	64	line slopes in LSP to LSF conversion
pow2.tab	table	33	lookup table in 2^x computation
acelp.h			prototypes for fixed codebook search
ld8k.h			prototypes and constants
typedef.h			type definitions

TABLE 13

Summary of encoder specific routines.

Filename	Description
acelp_co.c	Search fixed codebook
autocorr.c	Compute autocorrelation for LP analysis
az_lsp.c	compute LSPs from LP coefficients
cod_ld8k.c	encoder routine
convolve.c	convolution operation
corr_xy2.c	compute correlation terms for gain quantization
enc_lag3.c	encode adaptive codebook index
g_pitch.c	compute adaptive codebook gain
gainpred.c	gain predictor
int_lpc.c	interpolation of LSP
inter_3.c	fractional delay interpolation
lag_wind.c	lag-windowing
levinson.c	levinson recursion
lspenc.c	LSP encoding routine
lspgetq.c	LSP quantizer
lspgett.c	compute LSP quantizer distortion
lspgetw.c	compute LSP weights
lsplast.c	select LSP MA predictor
lsppre.c	pre-selection first LSP codebook
lspprev.c	LSP predictor routines
lspsel1.c	first stage LSP quantizer
lspsel2.c	second stage LSP quantizer
lspstab.c	stability test for LSP quantizer
pitch_fr.c	closed-loop pitch search
pitch_ol.c	open-loop pitch search
pre_proc.c	pre-processing (HP filtering and scaling)
pwf.c	computation of perceptual weighting coefficients
qua_gain.c	gain quantizer
qua_lsp.c	LSP quantizer
relspwe.c	LSP quantizer

TABLE 14

Summary of decoder specific routines.

Filename	Description
d_lsp.c	decode LP information
de_acelp.c	decode algebraic codebook
dec_gain.c	decode gains
dec_lag3.c	decode adaptive codebook index
dec_ld8k.c	decoder routine
lspdec.c	LSP decoding routine
post_pro.c	post processing (HP filtering and scaling)
pred_lt3.c	generation of adaptive codebook
pst.c	postfilter routines

TABLE 15

Summary of general routines.

Filename	Description
basicop2.c	basic operators
bits.c	bit manipulation routines
gainpred.c	gain predictor
int_lpc.c	interpolation of LSP
inter_3.c	fractional delay interpolation
lsp_az.c	compute LP from LSP coefficients
lsp_lsf.c	conversion between LSP and LSF
lsp_lsf2.c	high precision conversion between LSP and LSF
lspexp.c	expansion of LSP coefficients
lspstab.c	stability test for LSP quantizer
p_parity.c	compute pitch parity
pred_lt3.c	generation of adaptive codebook
random.c	random generator
residu.c	compute residual signal
syn_filt.c	synthesis filter
weight_a.c	bandwidth expansion LP coefficients

The invention claimed is:

1. A method for use in a speech processing system which includes a first portion comprising an adaptive codebook and corresponding adaptive codebook amplifier and a second portion comprising a fixed codebook coupled to a pitch filter, the pitch filter comprising a delay memory coupled to a pitch filter amplifier, the method comprising:

determining the pitch filter gain based on a measure of periodicity of a speech signal; and

amplifying samples of a signal in said pitch filter based on said determined pitch filter gain.

2. The method of claim 1 wherein the adaptive codebook gain is delayed for one subframe.

3. The method of claim 1 where the signal reflecting the adaptive codebook gain is delayed in time.

4. The method of claim 1 wherein the signal reflecting the adaptive codebook gain comprises values which are greater than or equal to a lower limit and less than or equal to an upper limit.

5. The method of claim 1 wherein the speech signal comprises a speech signal being encoded.

6. The method of claim 1 wherein the speech signal comprises a speech signal being synthesized.

7. A speech processing system comprising:

a first portion including an adaptive codebook and means for applying an adaptive codebook gain, and

a second portion including a fixed codebook, a pitch filter, wherein the pitch filter includes a means for applying a pitch filter gain,

and wherein the improvement comprises:

means for determining said pitch filter gain, based on a measure of periodicity of a speech signal.

8. The speech processing system of claim 7 wherein the signal reflecting the adaptive codebook gain is delayed for one subframe.

9. The speech processing system of claim 7 wherein the pitch filter gain equals a delayed adaptive codebook gain.

10. The speech processing of claim 7 wherein the pitch filter gain is limited to a range of values greater than or equal to 0.2 and less than or equal to 0.8 and, within said range, comprises a delayed adaptive codebook gain.

11. The speech processing system of claim 7 wherein the signal reflecting the adaptive codebook gain is limited to a range of values greater than or equal to 0.2 and less than or equal to 0.8 and, within said range, comprises an adaptive codebook gain.

12. The speech processing system of claim 7 wherein said first and second portions generate first and second output signals and wherein the system further comprises:

means for summing the first .and second output signals; and

a linear prediction filter, coupled the means for summing, for generating a speech signal in response to the summed first and second signals.

13. The speech processing system of claim 12 further comprising a post filter for filtering said speech signal generated by said linear prediction filter.

14. The speech processing system of claim 7 wherein the speech processing system is used in a speech encoder.

15. The speech processing system of claim 7 wherein the speech processing system is used in aspeech decoder.

16. The speech processing system of claim 5 wherein the means for determining comprises a memory for delaying a signal reflecting the adaptive codebook gain used in said first portion.

17. A method for determining a gain of a pitch filter for use in a speech processing system, the system including a first portion comprising an adaptive codebook and corresponding adaptive codebook amplifier and a second portion comprising a fixed codebook coupled to a pitch filter, the pitch filter comprising a delay memory coupled to a pitch filter amplifier for applying said determined gain, the speech processing system for processing a speech signal, the method comprising:

determining the pitch filter gain based on periodicity of the speech signal.

18. A method for use in a speech processing system which includes a first portion which comprises an adaptive codebook and corresponding adaptive codebook amplifier and a second portion which comprises a fixed codebook coupled to a pitch filter, the pitch filter. comprising a delay memory coupled to a pitch filter amplifier, the method comprising:

delaying the adaptive codebook gain;

determining the pitch filter gain to be equal to the delayed adaptive codebook gain, except when the adaptive codebook gain is either less than 0.2 or greater than 0.8., in which cases the pitch filter gain is set equal to 0.2 or 0.8, respectively; and

amplifying samples of a signal in said pitch filter based on said determined pitch filter gain.

19. A speech processing system comprising:

a first portion including an adaptive codebook and means for applying an adaptive codebook gain, and

a second portion including a fixed codebook, a pitch filter, means for applying a second gain, wherein the pitch filter includes a means for applying a pitch filter gain,

and wherein the improvement comprises:

means for determining said pitch filter gain, said means for determining including means for setting the pitch filter gain equal to an adaptive codebook gain, said signal gain is either less than 0.2 or greater than 0.8., in which cases-the pitch filter gain is set equal to 0.2 or 0.8, respectively.

* * * * *

EXHIBIT B

US005699485A

United States Patent [19]

Shoham

[11] **Patent Number: 5,699,485**

[45] **Date of Patent: Dec. 16, 1997**

[54] **PITCH DELAY MODIFICATION DURING FRAME ERASURES**

[75] Inventor: **Yair Shoham**, Watchung, N.J.

[73] Assignee: **Lucent Technologies Inc.**, Murray Hill, N.J.

[21] Appl. No.: **482,709**

[22] Filed: **Jun. 7, 1995**

[51] **Int. Cl.**^{6} **G10L 9/14**; H04B 1/10

[52] **U.S. Cl.** **395/2.32**; 395/2.28; 395/2.29; 395/2.31; 371/31; 341/94

[58] **Field of Search** 395/2.28, 2.29, 395/2.3, 2.31, 2.32; 371/31; 341/94

[56] **References Cited**

U.S. PATENT DOCUMENTS

5,305,332	4/1994	Ozawa	371/31
5,550,543	8/1996	Chen et al.	341/94

OTHER PUBLICATIONS

Brian Bryden, Gerald E. Seguin, Jean Conan, Vijay K. Bhargava, and Andre Brind'Amour, "Error Correction/Masking for Digital Voice Transmission Over the Land Mobile Satellite System", IEEE Transactions on Communication, vol. 37, No. 3, pp. 309–314, Mar. 1989.

Primary Examiner—Allen R. MacDonald
Assistant Examiner—Tālivaldis Ivars Smits
Attorney, Agent, or Firm—Thomas A. Restaino; Kenneth M. Brown

[57] **ABSTRACT**

In a speech decoder which experiences frame erasure, the pitch delay associated with the first of consecutive erased frames is incremented. The incremented value is used as the pitch delay for the second of consecutive erased frames. Pitch delay associated with the first of consecutive erased frames may correspond to the last correctly received pitch delay information from a speech encoder (associated with a non-erased frame), or it may itself be the result of an increment added to a still previous value of pitch delay (associated with a still previous erased frame).

5 Claims, 4 Drawing Sheets

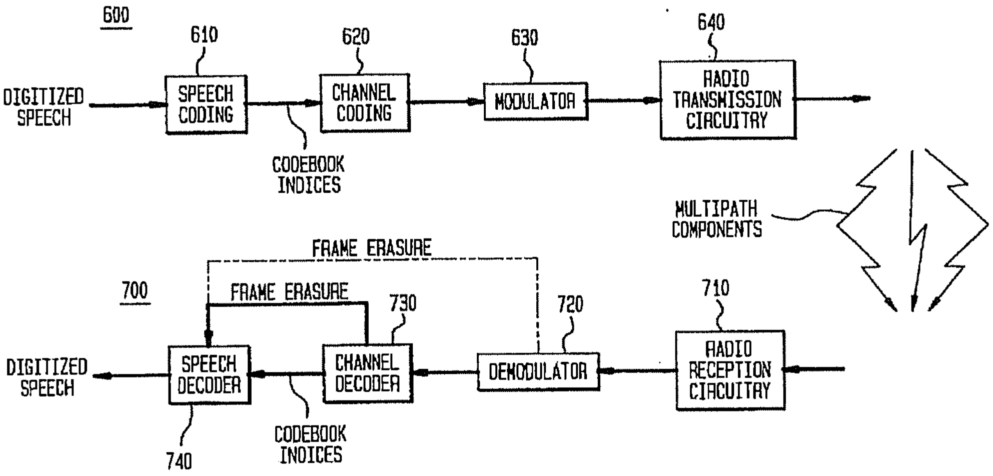


FIG. 1

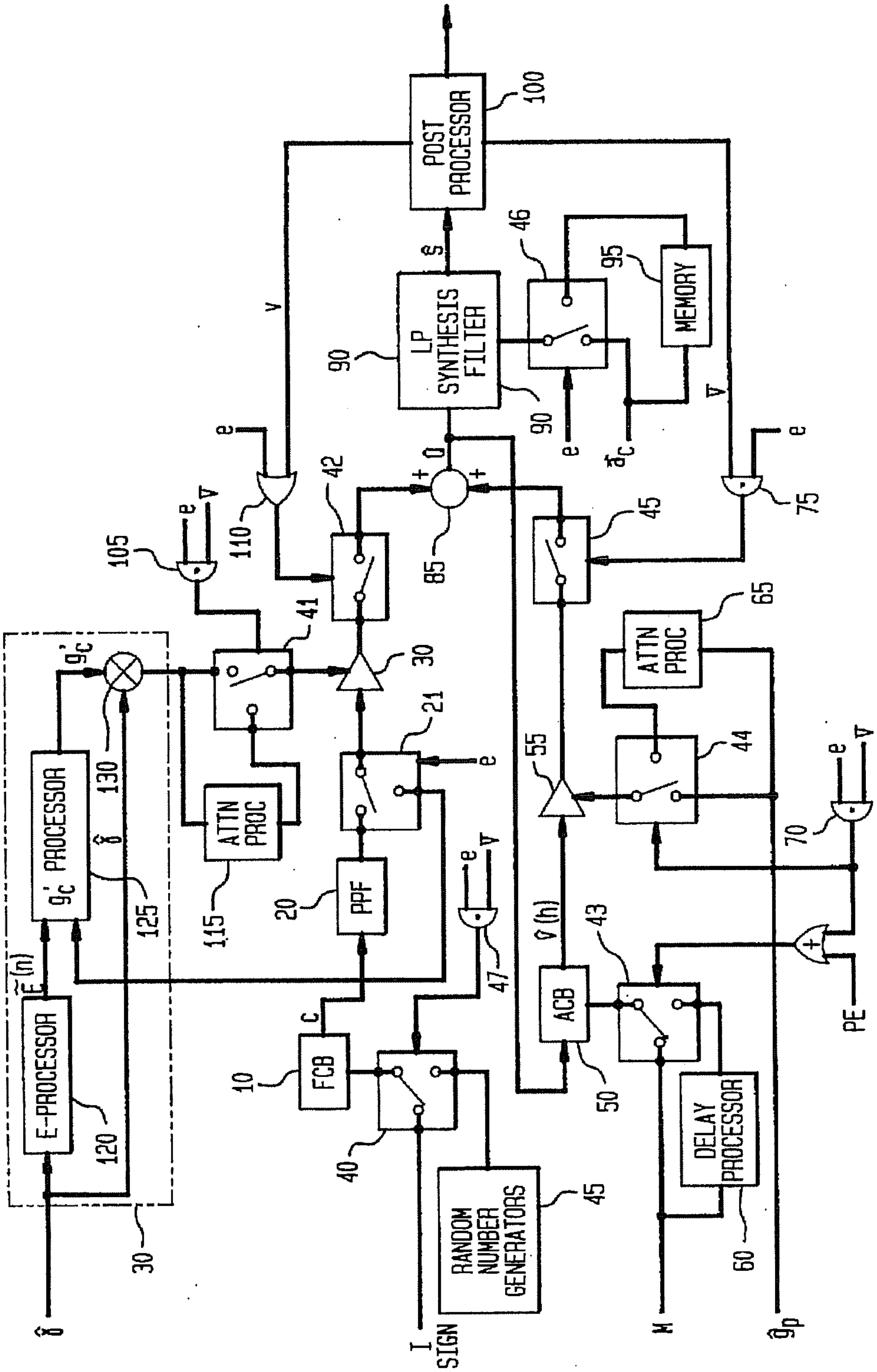

POST PROCESSOR
100
LP SYNTHESIS FILTER
90
90
s
v
46
95
MEMORY
e
$\hat{a}_c$
v
e
75
45
85
+
+
42
110
e
v
105
e
v
41
30
21
e
ATTN PROC
115
ATTN PROC
65
44
55
70
e
v
PPF
20
47
e
v
$\hat{v}(n)$
43
ACB
50
+
PE
FCB
10
c
40
RANDOM NUMBER GENERATORS
45
DELAY PROCESSOR
60
M
$\hat{g}_p$
I
SIGN
E-PROCESSOR
120
$\tilde{E}(n)$
g_c' PROCESSOR
$\hat{\gamma}$
125
130
g_c'
30
$\hat{\gamma}$

FIG. 2

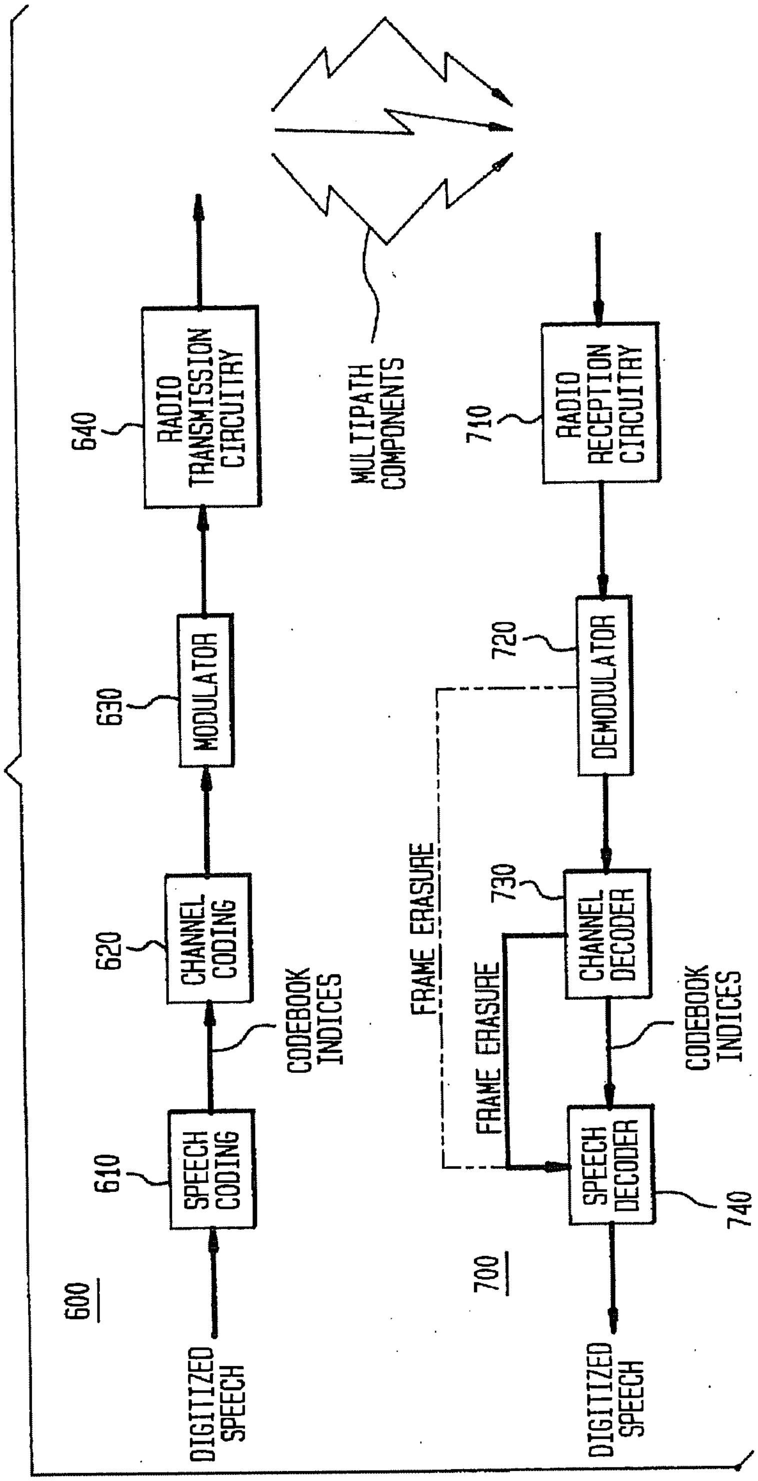

600
DIGITIZED SPEECH
610
SPEECH CODING
CODEBOOK INDICES
620
CHANNEL CODING
630
MODULATOR
640
RADIO TRANSMISSION CIRCUITRY
MULTIPATH COMPONENTS
700
DIGITIZED SPEECH
740
SPEECH DECODER
FRAME ERASURE
FRAME ERASURE
CODEBOOK INDICES
730
CHANNEL DECODER
720
DEMODULATOR
710
RADIO RECEPTION CIRCUITRY

FIG. 3

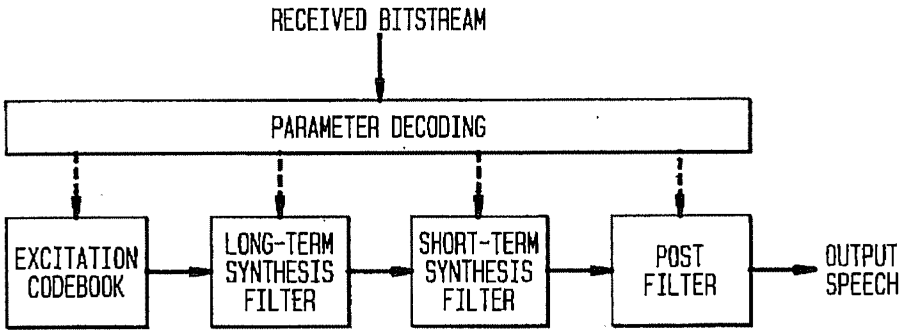

RECEIVED BITSTREAM
PARAMETER DECODING
EXCITATION CODEBOOK
LONG-TERM SYNTHESIS FILTER
SHORT-TERM SYNTHESIS FILTER
POST FILTER
OUTPUT SPEECH

FIG. 4

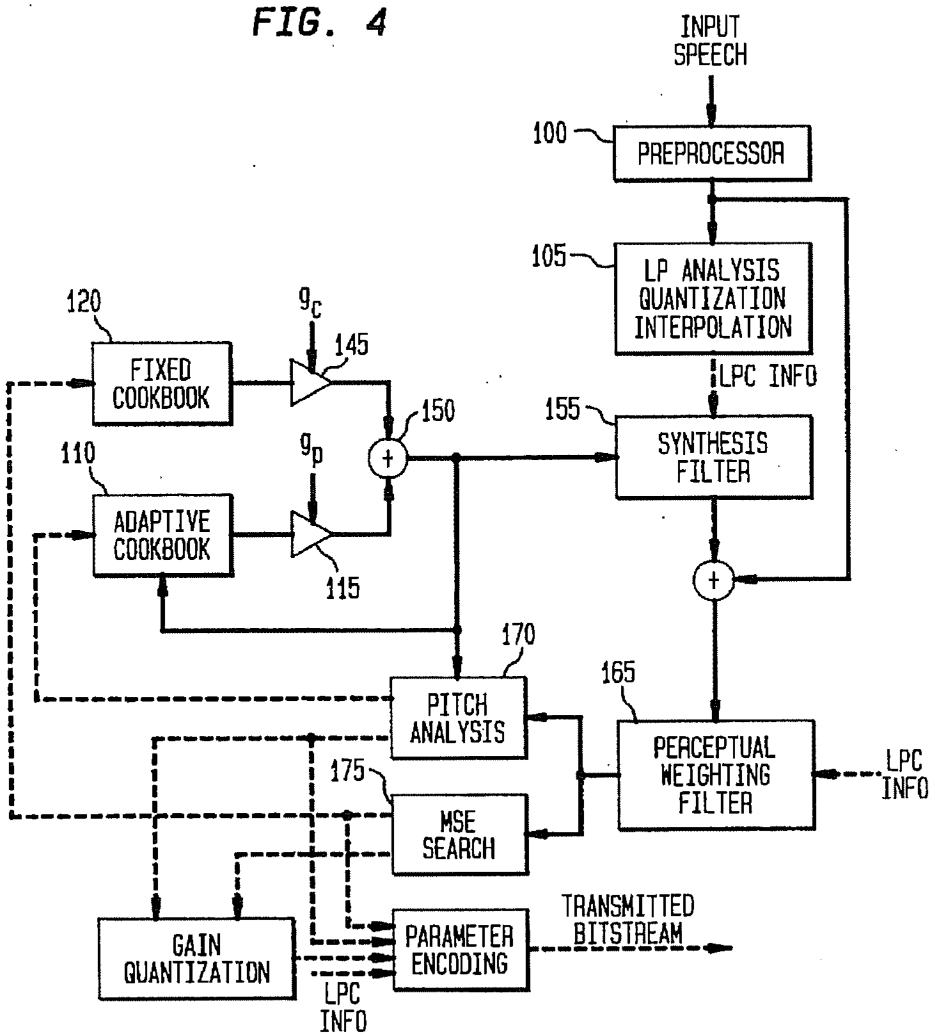

INPUT SPEECH
100
PREPROCESSOR
105
LP ANALYSIS QUANTIZATION INTERPOLATION
LPC INFO
120
FIXED COOKBOOK
g_c
145
150
155
SYNTHESIS FILTER
110
ADAPTIVE COOKBOOK
g_p
115
+
170
165
PITCH ANALYSIS
PERCEPTUAL WEIGHTING FILTER
LPC INFO
175
MSE SEARCH
GAIN QUANTIZATION
PARAMETER ENCODING
TRANSMITTED BITSTREAM
LPC INFO

FIG. 5

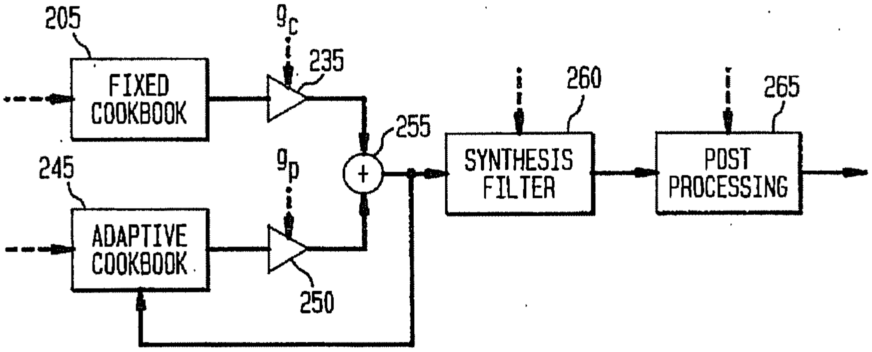
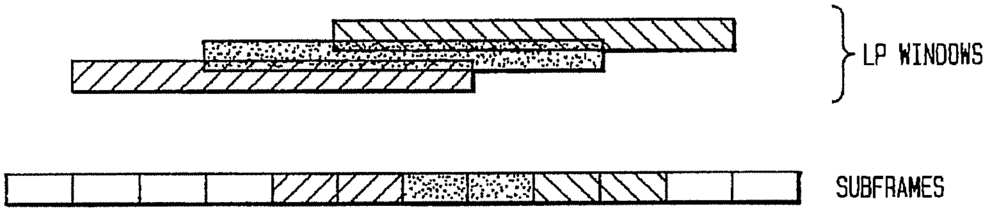

205
FIXED COOKBOOK
g_c
235
245
ADAPTIVE COOKBOOK
g_p
250
255
+
260
SYNTHESIS FILTER
265
POST PROCESSING

FIG. 6


LP WINDOWS
SUBFRAMES

PITCH DELAY MODIFICATION DURING FRAME ERASURES

CROSS-REFERENCE TO RELATED APPLICATION

This application is related to Application Ser. No. 08/482/715, entitled "Adaptive Codebook-Based Speech Compression System," filed on even date herewith, which is incorporated by reference as if set forth fully herein.

FIELD OF THE INVENTION

The present invention relates generally to speech coding arrangements for use in communication systems, and more particularly to the ways in which such speech coders function in the event of burst-like errors in transmission.

BACKGROUND OF THE INVENTION

Many communication systems, such as cellular telephone and personal communications systems, rely on wireless channels to communicate information. In the course of communicating such information, wireless communication channels can suffer from several sources of error, such as multipath fading. These error sources can cause, among other things, the problem of *frame erasure*. *Erasure* refers to the total loss or whole or partial corruption of a set of bits communicated to a receiver. A *frame* is a predetermined fixed number of bits which may be communicated as a block through a communication channel. A frame may therefore represent a time-segment of a speech signal.

If a frame of bits is totally lost, then the receiver has no bits to interpret. Under such circumstances, the receiver may produce a meaningless result. If a frame of received bits is corrupted and therefore unreliable, the receiver may produce a severely distorted result. In either case, the frame of bits may be thought of as "erased" in that the frame is unavailable or unusable by the receiver.

As the demand for wireless system capacity has increased, a need has arisen to make the best use of available wireless system bandwidth. One way to enhance the efficient use of system bandwidth is to employ a signal compression technique. For wireless systems which carry speech signals, speech compression (or *speech coding*) techniques may be employed for this purpose. Such speech coding techniques include analysis-by-synthesis speech coders, such as the well-known Code-Excited Linear Prediction (or CELP) speech coder.

The problem of packet loss in packet-switched networks employing speech coding arrangements is very similar to frame erasure in the wireless context. That is, due to packet loss, a speech decoder may either fail to receive a frame or receive a frame having a significant number of missing bits. In either case, the speech decoder is presented with the same essential problem—the need to synthesize speech despite the loss of compressed speech information. Both "frame erasure" and "packet loss" concern a communication channel (or network) problem which causes the loss of transmitted bits. For purposes of this description, the term "frame erasure" may be deemed to include "packet loss."

Among other things, CELP speech coders employ a codebook of *excitation signals* to encode an original speech signal. These excitation signals, scaled by an excitation gain, are used to "excite" filters which synthesize a speech signal (or some precursor to a speech signal) in response to the excitation. The synthesized speech signal is compared to the original speech signal. The codebook excitation signal is

identified which yields a synthesized speech signal which most closely matches the original signal. The identified excitation signal's *codebook index* and *gain* representation (which is often itself a gain codebook index) are then communicated to a CELP decoder (depending upon the type of CELP system, other types of information, such as linear prediction (LPC) filter coefficients, may be communicated as well). The decoder contains codebooks identical to those of the CELP coder. The decoder uses the transmitted indices to select an excitation signal and gain value. This selected scaled excitation signal is used to excite the decoder's LPC filter. Thus excited, the LPC filter of the decoder generates a decoded (or quantized) speech signal—the same speech signal which was previously determined to be closest to the original speech signal.

Some CELP systems also employ other components, such as a *periodicity model* (e.g., a *pitch-predictive filter* or an *adaptive codebook*). Such a model simulates the periodicity of voiced speech. In such CELP systems, parameters relating to these components must also be sent to the decoder. In the case of an adaptive codebook, signals representing a *pitch-period* (*delay*) and adaptive codebook gain must also be sent to the decoder so that the decoder can recreate the operation of the adaptive codebook in the speech synthesis process.

Wireless and other systems which employ speech coders may be more sensitive to the problem of frame erasure than those systems which do not compress speech. This sensitivity is due to the reduced redundancy of coded speech (compared to uncoded speech) making the possible loss of each transmitted bit more significant. In the context of a CELP speech coders experiencing frame erasure, excitation signal codebook indices and other signals representing speech in the frame may be either lost or substantially corrupted preventing proper synthesis of speech at the decoder. For example, because of the erased frame(s), the CELP decoder will not be able to reliably identify which entry in its codebook should be used to synthesize speech. As a result, speech coding system performance may degrade significantly.

Because frame erasure causes the loss of excitation signal codebook indicies, LPC coefficients, adaptive codebook delay information, and adaptive and fixed codebook gain information, normal techniques for synthesizing an excitation signal in a speech decoder are ineffective. Therefore, these normal techniques must be replaced by alternative measures.

SUMMARY OF THE INVENTION

The present invention addresses the problem of the lack of codebook gain information during frame erasure. In accordance with the present invention, a codebook-based speech decoder which fails to receive reliably at least a portion of a current frame of compressed speech information uses a codebook gain which is an attenuated version of a gain from a previous frame of speech.

An illustrative embodiment of the present invention is a speech decoder which includes a codebook memory and a signal amplifier. The memory and amplifier are use in generating a decoded speech signal based on compressed speech information. The compressed speech information includes a scale-factor for use by the amplifier in scaling a codebook vector. When a frame erasure occurs, a scale-factor corresponding to a previous frame of speech is attenuated and the attenuated scale factor is used to amplify the codebook vector corresponding to the current erased frame of speech. Specific details of an embodiment of the present invention are presented in section II.D. of the Detailed Description set forth below.

The present invention is applicable to both fixed and adaptive codebook processing, and also to systems which insert decoder systems or other elements (such as a pitch-predictive filter) between a codebook and its amplifier. See section II.B.1 of the Detailed Description for a discussion relating to the present invention.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. **1** presents a block diagram of a G.729 Draft decoder modified in accordance with the present invention.

FIG. **2** presents an illustrative wireless communication system employing the embodiment of the present invention presented in FIG. **1**.

FIG. **3** presents a block diagram of a conceptual G.729 CELP synthesis model.

FIG. **4** presents the signal flow at the G.729 CS-ACELP encoder.

FIG. **5** presents the signal flow at the G.729 CS-ACELP encoder.

FIG. **6** presents an illustration of windowing in LP anaylsis.

DETAILED DESCRIPTION

I. Introduction

The present invention concerns the operation of a speech coding system experiencing frame erasure—that is, the loss of a group of consecutive bits in the compressed bit-stream, which group is ordinarily used to synthesize speech. The description which follows concerns features of the present invention applied illustratively to an 8 kbit/s CELP speech coding system proposed to the ITU for adoption as its international standard G.729. For the convenience of the reader, a preliminary draft recommendation for the G.729 standard is provided in Section III. Sections III.3 and III.4 include detailed descriptions of the speech encoder and decoder, respectively. The illustrative embodiment of the present invention is directed to modifications of normal G.729 decoder operation, as detailed in G.729 Draft section 4.3. No modifications to the encoder are required to implement the present invention.

The applicability of the present invention to the proposed G.729 standard notwithstanding, those of ordinary skill in the art will appreciate that features of the present invention have applicability to other speech coding systems.

Knowledge of the erasure of one or more frames is an input signal, e, to the illustrative embodiment of the present invention. Such knowledge may be obtained in any of the conventional ways well-know in the art. For example, whole or partially corrupted frames may be detected through the use of a conventional error detection code. When a frame is determined to have been erased, e=1 and special procedures are initiated as described below. Otherwise, if not erased (e=0) normal procedures are used. Conventional error protection codes could be implemented as part of a conventional radio transmission/reception subsystem of a wireless communication system.

In addition to the application of the full set of remedial measures applied as the result of an erasure (e=1), the decoder employs a subset of these measures when a parity error is detected. A parity bit is computed based on the pitch delay index of the first of two subframes of a frame of coded speech. See Subsection III.3.7.1. This parity bit is computed by the decoder and checked against the parity bit received from the encoder. If the two parity bits are not the same, the delay index is said to be corrupted (PE=1, in the embodiment) and special processing of the pitch delay is invoked.

For clarity of explanation, the illustrative embodiment of the present invention is presented as comprising individual functional blocks. The functions these blocks represent may be provided through the use of either shared or dedicated hardware, including, but not limited to, hardware capable of executing software. For example, the blocks presented in FIG. **1** may be provided by a single shared processor. (Use of the term "processor" should not be construed to refer exclusively to hardware capable of executing software.)

Illustrative embodiments may comprise digital signal processor (DSP) hardware, such as the AT&T DSP16 or DSP32C, read-only memory (ROM) for storing software performing the operations discussed below, and random access memory (RAM) for storing DSP results. Very large scale integration (VLSI) hardware embodiments, as well as custom VLSI circuitry in combination with a general purpose DSP circuit, may also be provided.

II. An Illustrative Embodiment

FIG. **1** presents a block diagram of a G.729 Draft decoder modified in accordance with the present invention (FIG. 1 is a version of FIG. **5** (showing the signal flow at the G.729 CS-ACELP encoder) that has been augmented to more clearly illustrate features of the claimed invention). In normal operation (i.e., without experiencing frame erasure) the decoder operates in accordance with the description provided in Subsections III.4.1–III.4.2. During frame erasure, the operation of the embodiment of FIG. 1 is augmented by special processing to make up for the erasure of information from the encoder.

A. Normal Decoder Operation

The encoder described in Section III provides a frame of data representing compressed speech every 10 ms. The frame comprises 80 bits and is detailed in Tables 1 and 9 of Section III. Each 80-bit frame of compressed speech is sent over a communication channel to a decoder which synthesizes a speech (representing two subframes) signals based on the frame produced by the encoder. The channel over which the frames are communicated (not shown) may be of any type (such as conventional telephone networks, packet-based networks, cellular or wireless networks, ATM networks, etc.) and/or may comprise a storage medium (such as magnetic storage, semiconductor RAM or ROM, optical storage such as CD-ROM, etc.).

The illustrative decoder of FIG. **1** includes both an adaptive codebook (ACB) portion and a fixed codebook (FCB) portion. The ACB portion includes ACB **50** and a gain amplifier **55**. The FCB portion includes a FCB **10**, a pitch predictive filter (PPF) **20**, and gain amplifier **30**. The decoder decodes transmitted parameters (see Section III.4.1) and performs synthesis to obtain reconstructed speech.

The FCB **10** operates in response to an index, I, sent by the encoder. Index I is received through switch **40**. The FCB **10** generates a vector, c(n), of length equal to a subframe. See Section III.4.1.2. This vector is applied to the PPF **20**. PPF **20** operates to yield a vector for application to the FCB gain amplifier **30**. See Sections III.3.8 and III.4.1.3. The amplifier, which applies a gain, $\hat{g}_c$, from the channel, generates a scaled version of the vector produced by the PPF **20**. See Section III.4.1.3. The output signal of the amplifier **30** is supplied to summer **85** (through switch **42**).

The gain applied to the vector produced by PPF **20** is determined based on information provided by the encoder.

This information is communicated as codebook indices. The decoder receives these indicies and synthesizes a gain correction factor, $\hat{\gamma}$. See Section III.4.1.4. This gain correction factor, $\hat{\gamma}$, is supplied to code vector prediction energy (E-) processor **120**. E-processor **120** determines a value of the code vector predicted error energy, $\hat{R}$, in accordance with the following expression:

$$\hat{R}^{(n)}=20\log\hat{\gamma}[dB]$$

The value of $\hat{R}$ is stored in a processor buffer which holds the five most recent (successive) values of $\hat{R}$. $\hat{R}^{(n)}$ represents the predicted error energy of the fixed code vector at subframe n. The predicted mean-removed energy of the codevector is formed as a weighted sum of past values of $\hat{R}$:

$$\tilde{E}^{(n)}=\sum_{i=1}^{4}b_i\hat{R}^{(n-i)},$$

where b=[0.68 0.58 0.34 0.19] and where the past values of $\hat{R}$ are obtained from the buffer. This predicted energy is then output from processor **120** to a predicted gain processor **125**.

Processor **125** determines the actual energy of the code vector supplied by codebook **10**. This is done according to the following expression:

$$E=10\log\left(\frac{1}{40}\sum_{i=0}^{39}c_i^2\right),$$

where i indexes the samples of the vector. The predicted gain is then computed as follows:

$$g'_c=10^{(\tilde{E}(n)+\bar{E}-E)/20},$$

where $\bar{E}$ is the mean energy of the FCB (e.g., 30 dB)

Finally, the actual scale factor (or gain) is computed by multiplying the received gain correction factor, γ by the predicted gain, g'_c at multiplier **130**. This value is then supplied to amplifier **30** to scale the fixed codebook contribution provided by PPF **20**.

Also provided to the summer **85** is the output signal generated by the ACB portion of the decoder. The ACB portion comprises the ACB **50** which generates a excitation signal, v(n), of length equal to a subframe based on past excitation signals and the ACB pitch-period, M, received (through switch **43**) from encoder via the channel. See Subsection III.4.1.1. This vector is scaled by amplifier **250** based on gain factor, $\hat{g}_p$, received over the channel. This scaled vector is the output of the ACB portion.

Summer **85** generates an excitation signal, u(n), in response to signals from the FCB and ACB portions of the decoder. The excitation signal, u(n), is applied to an LPC synthesis filter **90** which synthesizes a speech signal based on LPC coefficients, a_i, received over the channel. See Subsection III.4.1.6.

Finally, the output of the LPC synthesis filter **90** is supplied to a post processor **100** which performs adaptive postfiltering (see Subsections III.4.2.1–III.4.2.4, high-pass filtering (see Subsections III.4.2.5), and up-scaling (see Subsections III.4.2.5).

B. Excitation Signal Synthesis During Frame Erasure

In the presence of frame erasures, the decoder of FIG. **1** does not receive reliable information (if it receives anything at all) from which an excitation signal, u(n), may be synthesized. As such, the decoder will not know which vector of signal samples should be extracted from codebook **10**, or what is the proper delay value to use for the adaptive codebook **50**. In this case, the decoder must obtain a *substitute* excitation signal for use in synthesizing a speech signal. The generation of a substitute excitation signal during periods of frame erasure is dependent on whether the erased frame is classified as *voiced* (periodic) or *unvoiced* (aperiodic). An indication of periodicity for the erased frame is obtained from the post processor **100**, which classifies each properly received frame as periodic or aperiodic. See Subsection III.4.2.1. The erased frame is taken to have the same periodicity classification as the previous frame processed by the postfilter. The binary signal representing periodicity, v, is determined according to postfilter variable g_{pit}. Signal v=1 if g_{pit}>0; else, v=0. As such, for example, if the last good frame was classified as periodic, v=1; otherwise v=0.

1. Erasure of Frames Representing Periodic Speech

For an erased frame (e=1) which is thought to have represented speech which is periodic (v=1), the contribution of the fixed codebook is set to zero. This is accomplished by switch **42** which switches states (in the direction of the arrow) from its normal (biased) operating position coupling amplifier **30** to summer **85** to a position which decouples the fixed codebook contribution from the excitation signal, u(n). This switching of state is accomplished in accordance with the control signal developed by AND-gate **110** (which tests for the condition that the frame is erased, e=1, and it was a periodic frame, v=1). On the other hand, the contribution of the adaptive codebook is maintained in its normal operating position by switch **45** (since e=1 but not_v=0).

The pitch delay, M, used by the adaptive codebook during an erased frame is determined by delay processor **60**. Delay processor **60** stores the most recently received pitch delay from the encoder. This value is overwritten with each successive pitch delay received. For the first erased frame following a "good" (correctly received) frame, delay processor **60** generates a value for M which is equal to the pitch delay of the last good frame (i.e., the previous frame). To avoid excessive periodicity, for each successive erased frame processor **60** increments the value of M by one (1). The processor **60** restricts the value of M to be less than or equal to 143 samples. Switch **43** effects the application of the pitch delay from processor **60** to adaptive codebook **50** by changing state from its normal operating position to its "voiced frame erasure" position in response to an indication of an erasure of a voiced frame (since e=1 and v=1).

The adaptive codebook gain is also synthesized in the event of an erasure of a voiced frame in accordance with the procedure discussed below in section C. Note that switch **44** operates identically to switch **43** in that it effects the application of a synthesized adaptive codebook gain by changing state from its normal operating position to its "voiced frame erasure" position.

2. Erasure of Frames Representing Aperiodic Speech

For an erased frame (e=1) which is thought to have represented speech which is aperiodic (v=0), the contribution of the adaptive codebook is set to zero. This is accomplished by switch **45** which switches states (in the direction of the arrow) from its normal (biased) operating position coupling amplifier **55** to summer **85** to a position which decouples the adaptive codebook contribution from the excitation signal, u(n). This switching of state is accomplished in accordance with the control signal developed by AND-gate **75** (which tests for the condition that the frame is erased, e=1, and it was an aperiodic frame, not_v=1). On the other hand, the contribution of the fixed codebook is maintained in its normal operating position by switch **42** (since e=1 but v=0).

The fixed codebook index, I, and codebook vector sign are not available do to the erasure. In order to synthesize a fixed codebook index and sign index from which a codebook vector, c(n), could be determined, a random number generator **45** is used. The output of the random number generator **45** is coupled to the fixed codebook **10** through switch **40**. Switch **40** is normally is a state which couples index I and sign information to the fixed codebook. However, gate **47** applies a control signal to the switch which causes the switch to change state when an erasure occurs of an aperiodic frame (e=1 and not_v=1).

The random number generator **45** employs the function:

$$seed=seed*31821+13849$$

to generate the fixed codebook index and sign. The initial seed value for the generator **45** is equal to 21845. For a given coder subframe, the codebook index is the 13 least significant bits of the random number. The random sign is the 4 least significant bits of the *next* random number. Thus the random number generator is run twice for each fixed codebook vector needed. Note that a noise vector could have been generated on a sample-by-sample basis rather than using the random number generator in combination with the FCB.

The fixed codebook gain is also synthesized in the event of an erasure of an aperiodic frame in accordance with the procedure discussed below in section D. Note that switch **41** operates identically to switch **40** in that it effects the application of a synthesized fixed codebook gain by changing state from its normal operating position to its "voiced frame erasure" position.

Since PPF **20** adds periodicity (when delay is less than a subframe), PPF **20** should not be used in the event of an erasure of an aperiodic frame. Therefore switch **21** selects either the output of FCB **10** when e=0 or the output of PPF **20** when e=1.

C. LPC Filter Coefficients for Erased Frames

The excitation signal, u(n), synthesized during an erased frame is applied to the LPC synthesis filter **90**. As with other components of the decoder which depend on data from the encoder, the LPC synthesis filter **90** must have substitute LPC coefficients, a_i, during erased frames. This is accomplished by repeating the LPC coefficients of the last good frame. LPC coefficients received from the encoder in a non-erased frame are stored by memory **95**. Newly received LPC coefficients overwrite previously received coefficients in memory **95**. Upon the occurrence of a frame erasure, the coefficients stored in memory **95** are supplied to the LPC synthesis filter via switch **46**. Switch **46** is normally biased to couple LPC coefficients received in a good frame to the filter **90**. However, in the event of an erased frame (e=1), the switch changes state (in the direction of the arrow) coupling memory **95** to the filter **90**.

D. Attenuation of Adaptive and Fixed Codebook Gains

As discussed above, both the adaptive and fixed codebooks **50, 10** have a corresponding gain amplifier **55, 30** which applies a scale factor to the codebook output signal. Ordinarily, the values of the scale factors for these amplifiers is supplied by the encoder. However, in the event of a frame erasure, the scale factor information is not available from the encoder. Therefore, the scale factor information must be synthesized.

For both the fixed and adaptive codebogks, the synthesis of the scale factor is accomplished by attenuation processors **65** and **115** which scale (or attenuate) the value of the scale factor used in the previous subframe. Thus, in the case of a frame erasure following a good frame, the value of the scale factor of the first subframe of the erased frame for use by the amplifier is the second scale factor from the good frame

multiplied by an attenuation factor. In the case of successive erased subframes, the later erased subframe (subframe n) uses the value of the scale factor from the former erased subframe (subframe n−1) multiplied by the attenuation factor. This technique is used no matter how many successive erased frames (and subframes) occur. Attenuation processors **65, 115** store each new scale factor, whether received in a good frame or synthesized for an erased frame, in the event that the next subframe will be en erased subframe.

Specifically, attenuation processor **115** synthesizes the fixed codebook gain, g_c, for erased subframe n in accordance with:

$$g_c^{(n)}=0.98\ g_c^{(n-1)}.$$

Attenuation processor **65** synthesizes the adaptive codebook gain, g_p, for erased subframe n in accordance with:

$$g_p^{(n)}=0.9\ g_p^{(n-1)}.$$

In addition, processor **65** limits (or clips) the value of the synthesized gain to be less than 0.9. The process of attenuating gains is performed to avoid undesired perceptual effects.

E. Attenuation of Gain Predictor Memory

As discussed above, there is a buffer which forms part of E-Processor **120** which stores the five most recent values of the prediction error energy. This buffer is used to predict a value for the predicted energy of the code vector from the fixed codebook.

However, due to frame erasure, there will be no information communicated to the decoder from the encoder from which new values of the prediction error energy. Therefore, such values will have to be synthesized. This synthesis is accomplished by E-processor **120** according to the following expression:

$$\hat{R}^{(n)}=\left(0.25\sum_{i=1}^{4}\hat{R}^{(n)}\right)-4.0.$$

Thus, a new value for $\hat{R}^{(n)}$ is computed as the average of the four previous values of $\hat{R}$ less 4 dB. The attenuation of the value of $\hat{R}$ is performed so as to ensure that once a good frame is received undesirable speech distortion is not created. The value of the synthesized R is limited not to fall below −14 dB.

F. An Illustrative Wireless System

As stated above, the present invention has application to wireless speech communication systems. FIG. **2** presents an illustrative wireless communication system employing an embodiment of the present invention. FIG. **2** includes a transmitter **600** and a receiver **700**. An illustrative embodiment of the transmitter **600** is a wireless base station. An illustrative embodiment of the receiver **700** is a mobile user terminal, such as a cellular or wireless telephone, or other personal communications system device. (Naturally, a wireless base station and user terminal may also include receiver and transmitter circuitry, respectively.) The transmitter **600** includes a speech coder **610**, which may be, for example, a coder according to Section III. The transmitter further includes a conventional channel coder **620** to provide error detection (or detection and correction) capability; a conventional modulator **630**; and conventional radio transmission circuitry; all well known in the art. Radio signals transmitted by transmitter **600** are received by receiver **700** through a transmission channel. Due to, for example, possible destructive interference of various multipath components of the transmitted signal, receiver **700** may be in a deep fade preventing the clear reception of transmitted bits. Under such circumstances, frame erasure may occur.

Receiver **700** includes conventional radio receiver circuitry **710**, conventional demodulator **720**, channel decoder **730**, and a speech decoder **740** in accordance with the present invention. Note that the channel decoder generates a frame erasure signal whenever the channel decoder determines the presence of a substantial number of bit errors (or unreceived bits). Alternatively (or in addition to a frame erasure signal from the channel decoder), demodulator **720** may provide a frame erasure signal to the decoder **740**.

G. Discussion

Although specific embodiments of this invention have been shown and described herein, it is to be understood that these embodiments are merely illustrative of the many possible specific arrangements which can be devised in application of the principles of the invention. Numerous and varied other arrangements can be devised in accordance with these principles by those of ordinary skill in the art without departing from the spirit and scope of the invention.

In addition, although the illustrative embodiment of present invention refers to codebook "amplifiers," it will be understood by those of ordinary skill in the art that this term encompasses the scaling of digital signals. Moreover, such scaling may be accomplished with scale factors (or gains) which are less than or equal to one (including negative values), as well as greater than one.

The following section of the detailed description contains the G.729 Draft. This document, at the time of the filing of the present application, is intended to be submitted to a standards body of The International Telecommunications Union (ITU), and provides a more complete description of an illustrative 8 kbit/s speech coding system which employs, inter alia, the principles of the present invention.

III.1 INTRODUCTION

This Recommendation contains the description of an algorithm for the coding of speech signals at 8 kbit/s using Conjugate-Structure-Algebraic-Code-Excited Linear-Predictive (CS-ACELP) coding.

This coder is designed to operate with a digital signal obtained by first performing telephone bandwidth filtering (ITU Rec. G.710) of the analog input signal, then sampling it at 8000 Hz, followed by conversion to 16 bit linear PCM for the input to the encoder. The output of the decoder should be converted back to an analog signal by similar means. Other input/output characteristics, such as those specified by ITU Rec. G.711 for 64 kbit/s PCM data, should be converted to 16 bit linear PCM before encoding, or from 16 bit linear PCM to the appropriate format after decoding. The bitstream from the encoder to the decoder is defined within this standard.

This Recommendation is organized as follows: Subsection III.2 gives a general outline of the CS-ACELP algorithm. In Subsections III.3 and III.4, the CS-ACELP encoder and decoder principles are discussed, respectively. Subsection III.5 describes the software that defines this coder in 16 bit fixed point arithmetic.

II.2 GENERAL DESCRIPTION OF THE CODER

The CS-ACELP coder is based on the code-excited linear-predictive (CE, LP) coding model. The coder operates on speech frames of 10 ms corresponding to 80 samples at a sampling rate of 8000 samples/sec. For every 10 msec frame, the speech signal is analyzed to extract the parameters of the CELP model (LP filter coefficients, adaptive and fixed codebook indices and gains). These parameters are encoded and transmitted. The bit allocation of the coder parameters is shown in Table 1. At the decoder, these parameters are used to retrieve the excitation and synthesis filter parameters. The speech is reconstructed by filtering this excitation through the LP synthesis filter, as is shown in FIG. 3. The short-term synthesis filter is based on a 10th order linear prediction (LP) filter. The long-term, or pitch synthesis filter is implemented using the so-called adaptive codebook approach for delays less than the subframe length. After computing the reconstructed speech, it is further enhanced by a postfilter.

TABLE 1

Bit allocation of the 8 kbit/s CS-ACELP algorithm (10 msec frame).

Parameter	Codeword	Subframe 1	Subframe 2	Total per frame
LSP	L0, L1, L2, L3			18
Adaptive codebook delay	P1, P2	8	5	13
Delay parity	P0	1		1
Fixed codebook index	C1, C2	13	13	26
Fixed codebook sign	S1, S2	4	4	8
Codebook gains (stage 1)	GA1, GA2	3	3	6
Codebook gains (stage 2)	GB1, GB2	4	4	8
Total				80

III.2.1 Encoder

The signal flow at the encoder is shown in FIG. 4. The input signal is high-pass filtered and scaled in the pre-processing block. The pre-processed signal serves as the input signal for all subsequent analysis. LP analysis is done once per 10 ms frame to compute the LP filter coefficients. These coefficients are converted to line spectrum pairs (LSP) and quantized using predictive two-stage vector quantization (VQ) with 18 bits. The excitation sequence is chosen by using an analysis-by-synthesis search procedure in which the error between the original and synthesized speech is minimized according to a perceptually weighted distortion measure. This is done by filtering the error signal with a perceptual weighting filter, whose coefficients are derived from the unquantized LP filter. The amount of perceptual weighting is made adaptive to improve the performance for input signals with a flat frequency-response.

The excitation parameters (fixed and adaptive codebook parameters) are determined per subframe of 5 ms (40 samples) each. The quantized and unquantized LP filter coefficients are used for the second subframe, while in the first subframe interpolated LP filter coefficients are used (both quantized and unquantized). An open-loop pitch delay is estimated once per 10 ms frame based on the perceptually weighted speech signal. Then the following operations are repeated for each subframe. The target signal z(n) is computed by filtering the LP residual through the weighted synthesis filter $W(z)/\hat{A}(z)$. The initial states of these filters are updated by filtering the error between LP residual and excitation. This is equivalent to the common approach of subtracting the zero-input response of the weighted synthesis filter from the weighted speech signal. The impulse response, h(n), of the weighted synthesis filter is computed. Closed-loop pitch analysis is then done (to find the adaptive codebook delay and gain), using the target x(n) and impulse response h(n), by searching around the value of the open-loop pitch delay. A fractional pitch delay with ⅓ resolution is used. The pitch delay is encoded with 8 bits in the first subframe and differentially encoded with 5 bits in the second subframe. The target signal x(n) is updated by removing the adaptive codebook contribution (filtered adaptive codevector), and this new target, $x_2(n)$, is used in the fixed algebraic codebook search (to find the optimum excitation). An algebraic codebook with 17 bits is used for the fixed codebook excitation. The gains of the adaptive and fixed codebook are vector quantized with 7 bits, (with MA prediction applied to the fixed codebook gain). Finally, the filter memories are updated using the determined excitation signal.

II.2.2 Decoder

The signal flow at the decoder is shown in FIG. 5. First, the parameters indices are extracted from the received bitstream. These indices are decoded to obtain the coder parameters corresponding to a 10 ms speech frame. These parameters are the LSP coefficients, the 2 fractional pitch delays, the 2 fixed codebook vectors, and the 2 sets of adaptive and fixed codebook gains. The LSP coefficients are interpolated and converted to LP filter coefficients for each subframe. Then, for each 40-sample subframe the following steps are done:

- the excitation is constructed by adding the adaptive and fixed codebook vectors scaled by their respective gains,
- the speech is reconstructed by filtering the excitation through the LP synthesis filter,
- the reconstructed speech signal is passed through a post-processing stage, which comprises of an adaptive postfilter based on the long-term and short-term synthesis filters, followed by a high-pass filter and scaling operation.

III.2.3 Delay

This coder encodes speech and other audio signals with 10 ms frames. In addition, there is a look-ahead of 5 ms, resulting in a total algorithmic delay of 15 ms. All additional delays in a practical implementation of this coder are due to:

- processing time needed for encoding and decoding operations,
- transmission time on the communication link,
- multiplexing delay when combining audio data with other data.

III.2.4 Speech Coder Description

The description of the speech coding algorithm of this Recommendation is made in terms of bit-exact, fixed-point mathematical operations. The ANSI C code indicated in Subsection III.5, which constitutes an integral part of this Recommendation, reflects this bit-exact, fixed-point descriptive approach. The mathematical descriptions of the encoder (Subsection III.3), and decoder (Subsection III.4), can be implemented in several other fashions, possibly leading to a codec implementation not complying with this Recommendation. Therefore, the algorithm description of the C code of Subsection III.5 shall take precedence over the mathematical descriptions of Subsections III.3 and III.4 whenever discrepancies are found. A non-exhaustive set of test sequences which can be used in conjunction with the C code are available from the ITU.

III.2.5 Notational Conventions

Throughout this document it is tried to maintain the following notational conventions.

- Codebooks are denoted by caligraphic characters (e.g. C).
- Time signals are denoted by the symbol and the sample time index between parenthesis (e.g. s(n)). The symbol n is used as sample instant index.
- Superscript time indices (e.g $g^{(m)}$) refer to that variable corresponding to subframe m.
- Superscripts identify a particular element in a coefficient array.
- A identifies a quantized version of a parameter.
- Range notations are done using square brackets, where the boundaries are included (e.g. [0.6, 0.9]).
- log denotes a logarithm with base 10.

Table 2 lists the most relevant symbols used throughout this document. A glossary of the most relevant signals is given in Table 3. Table 4 summarizes relevant variables and their dimension. Constant parameters are listed in Table 5. The acronyms used in this Recommendation are summarized in Table 6.

TABLE 2

Glossary of symbols.

Name	Reference	Description
1/A(z)	Eq. (2)	LP synthesis filter
$H_{h1}(z)$	Eq. (1)	input high-pass filter
$H_p(z)$	Eq. (77)	pitch postfilter
$H_f(z)$	Eq. (83)	short-term postfilter
$H_t(z)$	Eq. (85)	tilt-compensation filter
$H_{h2}(z)$	Eq. (90)	output high-pass filter
P(z)	Eq. (46)	pitch filter
W(z)	Eq. (27)	weighting filter

TABLE 3

Glossary of signals.

Name	Description
h(n)	impulse response of weighting and synthesis filters
r(k)	auto-correlation sequence
r'(k)	modified auto-correlation sequence
R(k)	correlation sequence
sw(n)	weighted speech signal
s(n)	speech signal
s'(n)	windowed speech signal
sf(n)	postfiltered output
sf'(n)	gain-scaled postfiltered output
$\hat{s}(n)$	reconstructed speech signal
r(n)	residual signal
x(n)	target signal
$x_2(n)$	second target signal
v(n)	adaptive codebook contribution
c(n)	fixed codebook contribution
y(n)	v(n) * h(n)
z(n)	c(n) * h(n)
u(n)	excitation to LP synthesis filter
d(n)	correlation between target signal and h(n)
ew(n)	error signal

TABLE 4

Glossary of variables.

Name	Size	Description
g_p	1	adaptive codebook gain
g_c	1	fixed codebook gain
g_c	1	modified gain for pitch postfilter
g_{pit}	1	pitch gain for pitch postfilter

TABLE 4-continued

Glossary of variables.

Name	Size	Description
g_f	1	gain term short-term postfilter
g_t	1	gain term tilt postfilter
T_{op}	1	open-loop pitch delay
a_i	10	LP coefficients
k_i	10	reflection coefficients
o_1	2	LAR coefficients
w_i	10	LSF normalized frequencies
q_i	10	LSP coefficients
r(k)	11	correlation coefficients
w_i	10	LSP weighting coefficients
l_i	10	LSP quantizer output

TABLE 5

Glossary of constants.

Name	Value	Description
f_s	8000	sampling frequency
f_0	60	bandwidth expansion
γ_1	0.94/0.98	weight factor perceptual weighting filter
γ_2	0.60/[0.4–0.7]	weight factor perceptual weighting filter
γ_n	0.55	weight factor post filter
γ_d	0.70	weight factor post filter
γ_p	0.50	weight factor pitch post filter
γ_t	0.90/0.2	weight factor tilt post filter
C	Table 7	fixed (algebraic) codebook
L0	Section 3.2.4	moving average predictor codebook
L1	Section 3.2.4	First stage LSP codebook
L2	Section 3.2.4	Second stage LSP codebook (low part)
L3	Section 3.2.4	Second stage LSP codebook (high part)
GA	Section 3.9	First stage gain codebook
GB	Section 3.9	Second stage gain codebook
w_{lag}	Eq. (6)	correlation lag window
w_{lp}	Eq. (3)	LPC analysis window

TABLE 6

Glossary of acronyms.

Acronym	Description
CELP	code-excited linear-prediction
MA	moving average
MSB	most significant bit
LP	linear prediction
LSP	line spectral pair
LSF	line spectral frequency
VQ	vector quantization

III.3 FUNCTIONAL DESCRIPTION OF THE ENCODER

In this section we describe the different functions of the encoder represented in the blocks of FIG. 3.

III.3.1 Pre-Processing

As stated in Subsection III.2, the input to the speech encoder is assumed to be a 16 bit PCM signal. Two pre-processing functions are applied before the encoding process: 1) signal scaling, and 2) high-pass filtering.

The scaling consists of dividing the input by a factor 2 to reduce the possibility of overflows in the fixed-point implementation. The high-pass filter serves as a precaution against undesired low-frequency components. A second order pole/zero filter with a cutoff frequency of 140 Hz is used. Both the scaling and high-pass filtering are combined by dividing the coefficients at the numerator of this filter by 2. The resulting filter is given by

$$H_{h1}(z)=\frac{0.46363718-0.92724705z^{-1}+0.46363718z^{-2}}{1-1.9059465z^{-1}+0.9114024z^{-2}}. \quad (1)$$

The input signal filtered through $H_{h1}(z)$ is referred to as s(n), and will be used in all subsequent coder operations.

III.3.2 Linear Prediction Analysis and Quantization

The short-term analysis and synthesis filters are based on 10th order linear prediction (LP) filters. The LP synthesis filter is defined as

$$\frac{1}{\hat{A}(z)}=\frac{1}{1+\sum_{i=1}^{10}\hat{a}_i z^{-i}}, \quad (2)$$

where $\hat{a}_i$, i=1, . . . , 10, are the (quantired) linear prediction (LP) coefficients. Short-term prediction, or linear prediction analysis is performed once per speech frame using the autocorrelation approach with a 30 ms asymmetric window. Every 80 samples (10 ms), the autocorrelation coefficients of windowed speech are computed and converted to the LP coefficients using the Levinson algorithm. Then the LP coefficients are transformed to the LSP domain for quantization and interpolation purposes. The interpolated quantized and unquantized filters are converted back to the LP filter coefficients (to construct the synthesis and weighting filters at each subframe).

III.3.2.1 Windowing and Autocorrelation Computation

The LP analysis window consists of two parts: the first part is half a Hamming window and the second part is a quarter of a cosine function cycle. The window is given by:

$$w_{lp}(n)=\begin{cases}0.54-0.46\cos\left(\frac{2\pi n}{399}\right), & n=0,\ldots,199,\\ \cos\left(\frac{2\pi(n-200)}{159}\right), & n=200,\ldots,239.\end{cases} \quad (3)$$

There is a 5 ms lookahead in the LP analysis which means that 40 samples are needed from the future speech frame. This translates into an extra delay of 5 ms at the encoder stage. The LP analysis window applies to 120 samples from past speech frames, 80 samples from the present speech frame, and 40 samples from the future frame. The windowing in LP analysis is illustrated in FIG. 6.

The autocorrelation coefficients of the windowed speech

$$s'(n)=w_{lp}(n)s(n),\ n=0,\ldots,239, \quad (4)$$

are computed by

$$r(k)=\sum_{n=k}^{239}s'(n)s'(n-k),\quad k=0,\ldots,10, \quad (5)$$

To avoid arithmetic problems for low-level input signals the value of r(0) has a lower boundary of r(0)=1.0. A 60 Hz bandwidth expansion is applied, by multiplying the autocorrelaion coefficients with

$$w_{lag}(k)=\exp\left[-\frac{1}{2}\left(\frac{2\pi f_0 k}{f_s}\right)^2\right],\quad k=1,\ldots,10, \quad (6)$$

where f_0=60 Hz is the bandwidth expansion and f_s=8000 Hz is the sampling frequency. Further, r(0) is multiplied by the white noise correction factor 1.0001, which is equivMent to adding a noise floor at −40 dB.

III.3.2.2 Levinson-Durbin Algorithm

The modified autocorrelation coefficients

$$r'(0)=1.0001\ r(0)$$

$$r'(k)=w_{lag}(k)r(k),\ k=1, \ldots ,10 \tag{7}$$

are used to obtain the LP filter coefficients a_i, i=1, . . . , 10, by solving the set of equations

$$\sum_{i=1}^{10} a_i r'(|i-k|) = -r'(k), \quad k=1, \ldots, 10. \tag{8}$$

The set of equations in (8) is solved using the Levinson-Durbin algorithm. This algorithm uses the following recursion:

```
$E(0) = r'(0)$
for $i = 1$ to 10
    $a_0^{(i-1)} = 1$
    $k_i = -\left[\sum_{j=0}^{i-1} a_j^{(i-1)} r'(i-j)\right] / E(i-1)$
    $a_i^{(i)} = k_i$
    for $j = 1$ to $i-1$
        $a_j^{(i)} = a_j^{(i-1)} + k_i a_{i-j}^{(i-1)}$
    end
    $E(i) = (1-k_i^2)E(i-1)$,      if $E(i) < 0$ then $E(i) = 0.01$
end
```

The final solution is given as $a_j=a_j^{(10)}$, j=1, . . . , 10.

III.3.2.3 LP to LSP Conversion

The LP filter coefficients a_i, i=1, . . . , 10 are converted to the line spectral pair (LSP) representation for quantization and interpolation purposes. For a 10th order LP filter, the LSP coefficients are defined as the roots of the sum and difference polynomials

$$F'_1(z)=A(z)+z^{-11}A(z^{-1}), \tag{9}$$

and

$$F'_2(z)=A(z)-z^{-11}A(z^{-1}), \tag{10}$$

respectively. The polynomial $F'_1(z)$ is symmetric, and $F'_2(z)$ is antisymmetric. It can be proven that all roots of these polynomials are on the unit circle and they alternate each other. $F'_1(z)$ has a root z=−1 (ω=π) and $F'_2(z)$ has a root z=1 (ω=0). To eliminate these two roots, we define the new polynomials

$$F_1(z)=F'_1(z)/(1+z^{-1}), \tag{11}$$

and

$$F_2(z)=F'_2(z)/(1-z^{-1}). \tag{12}$$

Each polynomial has 5 conjugate roots on the unit circle $(e^{\pm j\omega i})$, therefore, the polynomials can be written as

$$F_1(z) = \prod_{i=1,3,\ldots,9} (1-2q_i z^{-1}+z^{-2}) \tag{13}$$

and

-continued

$$F_2(z) = \prod_{i=2,4,\ldots,10} (1-2q_i z^{-1}+z^{-2}), \tag{14}$$

where $q_i=\cos(\omega_i)$ with ω_i being the line spectral frequencies (LSF) and they satisfy the ordering property $0<\omega_1<\omega_2<. . .<\omega_{10}<\pi$. We refer to q_i as the LSP coefficients in the cosine domain.

Since both polynomials $F_1(z)$ and $F_2(z)$ are symmetric only the first 5 coefficients of each polynomial need to be computed. The coefficients of these polynomials are found by the recursive relations

$$f_1(i+1)=a_{i+1}+a_{10-i}-f_1(i),\ i=0, \ldots ,4,$$

$$f_2(i+1)=a_{i+1}-a_{10-i}+f_2(i),\ i=0, \ldots ,4, \tag{15}$$

where $f_1(0)=f_2(0)=1.0$. The LSP coefficients are found by evaluating the polynomials $F_1(z)$ and $F_2(z)$ at 60 points equally spaced between 0 and π and checking for sign changes. A sign change signifies the existence of a root and the sign change interval is then divided 4 times to better track the root. The Chebyshev polynomials are used to evaluate $F_1(z)$ and $F_2(z)$. In this method the roots are found directly in the cosine domain $\{q_i\}$. The polynomials $F_1(z)$ or $F_2(z)$, evaluated at $z=e^{jw}$, can be written as

$$F(\omega)=2e^{-j5\omega}C(x), \tag{16}$$

with

$$C(x)=T_5(x)+f(1)T_4(x)+f(2)T_3(x)+f(3)T_2(x)+f(4)T_1(x)+f(5)/2, \tag{17}$$

where $T_m(x)=\cos(m\omega)$ is the mth order Chebyshev polynomial, and f(i), i=1, . . . ,5, are the coefficients of either $F_1(z)$ or $F_2(z)$, computed using the equations in (15). The polynomial C(x) is evaluated at a certain value of x=cos(ω) using the recursire relation:

```
for $k = 4$ downto 1
    $b_k = 2xb_{k+1} - b_{k+2} + f(5-k)$
end
$C(x)\ xb_1 - b_2 + f(5)/2$
```

with initial values $b_5=1$ and $b_6=0$.

III.3.2.4 Quantization of the LSP Coefficients

The LP filter coefficients are quantized using the LSP representation in the frequency domain; that is

$$\omega_i=\arccos(q_i),\ i=1, \ldots ,10, \tag{18}$$

where ω_i are the line spectral frequencies (LSF) in the normalized frequency domain [0, π]. A switched 4th order MA prediction is used to predict the current set of LSF coefficients. The difference between the computed and predicted set of coefficients is quantized using a two-stage vector quantizer. The first stage is a 10-dimensional VQ using codebook L1 with 128 entries (7 bits). The second stage is a 10 bit VQ which has been implemented as a split VQ using two 5-dimensional codebooks, L2 and L3 containing 32 entries (5 bits) each.

To explain the quantization process, it is convenient to first describe the decoding process. Each coefficient is obtained from the sum of 2 codebooks:

$$l_i = \begin{cases} \mathcal{L}1_i(L1) + \mathcal{L}2_i(L2) & i=1, \ldots, 5, \\ \mathcal{L}1_i(L1) + \mathcal{L}3_{(i-5)}(L3) & i=6, \ldots, 10, \end{cases} \tag{19}$$

where L1, L2, and L3 are the codebook indices. To avoid sharp resonances in the quantized LP synthesis filters, the coefficients l_i are arranged such that adjacent coefficients have a minimum distance of J. The rearrangement routine is shown below:

```
for $i = 2, \ldots, 10$
    if $(l_{i-1} > l_i - J)$
        $l_{i-1} = (l_i + l_{i-1} - J)/2$
        $l_i = (l_i + l_{i-1} + J)/2$
    end
end
```

This rearrangement process is executed twice. First with a value of J=0.0001; then with a value of J=0.000095.

After this rearrangement process, the quantized LSF coefficients $\overline{\omega}_i^{(m)}$ for the current frame n, are obtained from the weighted sum of previous quantizer outputs $l^{(m-k)}$), and the current quantizer output $l^{(m)}$

$$\hat{\omega}_i^{(m)} = \left(1 - \sum_{k=1}^{4} m_i^k\right) l_i^{(m)} + \sum_{k=1}^{4} m_i^k l_i^{(m-k)}, \quad i = 1, \ldots, 10, \tag{20}$$

where m_i^k are the coefficients of the switched MA predictor. Which MA predictor to use is defined by a separate bit **L0**. At startup the initial values of $l_i^{(k)}$ are given by $l_i = i\pi/11$ for all k<0.

After computing $\overline{\omega}_i$, the corresponding filter is checked for stability. This is done as follows:

1. Order the coefficient $\overline{\omega}_i$ in increasing value,
2. If $\overline{\omega}_1 < 0.005$ then $\overline{\omega}_i = 0.005$,
3. If $\overline{\omega}_{i+1} - \overline{\omega}_i < 0.0001$, then $\overline{\omega}_{i+1} = \overline{\omega}_i + 0.0001$ i=1, . . . ,9,
4. If $\overline{\omega}_{10} > 3.135$ then $\overline{\omega}_{10} = 3.135$.

The procedure for encoding the LSF parameters can be outlined as follows. For each of the two MA predictors the best approximation to the current LSF vector has to be found. The best approximation is defined as the one that minimizes a weighted mean-squared error

$$E_{LPC} = \sum_{i=1}^{10} w_i(\omega_i - \hat{\omega}_i)^2. \tag{21}$$

The weights ω_i are made adaptive as a function of the unquantized LSF coefficients,

$$w_1 = \begin{cases} 1.0 & \text{if } \omega_2 - 0.04\pi - 1 > 0, \\ 10(\omega_2 - 0.04\pi - 1)^2 + 1 & \text{otherwise} \end{cases} \tag{22}$$

$$w_i\, 2 \leqq i \leqq 9 = \begin{cases} 1.0 & \text{if } \omega_{i+1} - \omega_{i-1} - 1 > 0, \\ 10(\omega_{i+1} - \omega_{i-1} - 1)^2 + 1 & \text{otherwise} \end{cases}$$

$$w_{10} = \begin{cases} 1.0 & \text{if } -\omega_9 + 0.92\pi - 1 > 0, \\ 10(-\omega_9 + 0.92\pi - 1)^2 + 1 & \text{otherwise} \end{cases}$$

In addition, the weights ω_5 and ω_6 are multiplied by 1.2 each.

The vector to be quantized for the current frame is obtained from

$$r_i = \left[\hat{\omega}_i^{(m)} - \sum_{k=1}^{4} m_i^k l_i^{(m-k)}\right] / \left(1 - \sum_{k=1}^{4} m_i^k\right), \quad i = 1, \ldots, 10. \tag{23}$$

The first codebook **L1** is searched and the entry **L1** that minimizes the (unweighted) meansquared error is selected. This is followed by a search of the second codebook L2, which defines the lower part of the second stage. For each possible candidate, the partial vector $\overline{\omega}_i$=1, . . . ,5 is reconstructed using Eq. (20), and rearranged to guarantee a minimum distance of 0.0001. The vector with index L2 which after addition to the first stage candidate and rearranging, approximates the lower part of the corresponding target best in the weighted MSE sense is selected. Using the selected first stage vector **L1** and the lower part of the second stage (L2), the higher part of the second stage is searched from codebook L3. Again the rearrangement procedure is used to guarantee a minimum distance of 0.0001. The vector L3 that minimizes the overall weighted MSE is selected.

This process is done for each of the two MA predictors defined by **L0**, and the MA predictor **L0** that produces the lowest weighted MSE is selected.

III.3.2.5 Interpolation of the LSP Coefficients

The quantized (and unquantized) LP coefficients are used for the second subframe. For the first subframe, the quantized (and unquantized) LP coefficients are obtained from linear interpolation of the corresponding parameters in the adjacent subframes. The interpolation is done on the LSP coefficients in the q domain. Let $q_i^{(m)}$ be the LSP coefficients at the 2nd subframe of frame m, and $q_i^{(m-1)}$ the LSP coefficients at the 2nd subframe of the past frame (m−1). The (unquantized) interpolated LSP coefficients in each of the 2 subframes are given by

$$\begin{array}{lll} \text{Subframe 1:} \quad q1_i = & 0.5q_i^{(m-1)} + 0.5q_i^{(m)}, & i = 1, \ldots, 10, \\ \text{Subframe 2:} \quad q2_i = & q_i^m & i = 1, \ldots, 10. \end{array} \tag{24}$$

The same interpolation procedure is used for the interpolation of the quantized LSP coeffidents by substituting q_i by $\overline{q}_i$ in Eq. (24).

III.3.2.6 LSP to LP Conversion

Once the LSP coefficients are quantized and interpolated, they are converted back to LP coefficients $\{a_i\}$. The conversion to the LP domain is done as follows. The coefficients of $F_1(z)$ and $F_2(z)$ are found by expanding Eqs. (13) and (14) knowing the quantized and interpolated LSP coefficients. The following recursive relation is used to compute $f_1(i)$, i=1, . . . ,5, from q_i

```
for $i = 1$ to 5
    $f_1(i) = -2q_{2i-1}f_1(i-1) + 2f_1(i-2)$
    for $j = i - 1$ downto 1
        $f_1(j) = f_1(j) - 2q_{2i-1}f_1(j-1) + f_1(j-2)$
    end
end
```

with initial values $f_1(0)=1$ and $f_1(-1)=0$. The coefficients $f_2(i)$ are computed similarly by replacing q_{2i-1} by q_{2i}.

Once the coefficients $f_1(i)$ and $f_2(i)$ are found, $F_1(z)$ and $F_2(z)$ are multiplied by $1+z^{-1}$ and $1-z^{-1}$ respectively, to obtain $F'_1(z)$ and $F'_2(z)$; that is

$$\begin{array}{ll} f_1'(i) = f_1(i) + f_1(i-1), & i = 1, \ldots, 5, \\ f_2'(i) = f_2(i) - f_2(i-1), & i = 1, \ldots, 5. \end{array} \tag{25}$$

Finally the LP coefficients are found by

$$a_i = \begin{cases} 0.5f_1'(i) + 0.5f_2'(i), & i = 1, \ldots, 5, \\ 0.5f_1'(i-5) - 0.5f_2'(i-5), & i = 1, \ldots, 10. \end{cases} \tag{26}$$

This is directly derived from the relation $A(z)=(F'_1(z)+F_2(z))/2$, and because $F'_1(z)$ and $F'_2(z)$ are symmetric and antisymmetric polynomials, respectively.

III.3.3 Perceptual Weighting

The perceptual weighting filter is based on the unquantized LP filter coefficients and is given by

$$W(z)=\frac{A(z/\gamma_1)}{A(z/\gamma_2)}=\frac{1+\sum_{i=1}^{10}\gamma_1^{i}a_iz^{-i}}{1+\sum_{i=1}^{10}\gamma_2^{i}a_iz^{-i}}. \tag{27}$$

The values of γ_1 and γ_2 determine the frequency response of the filter W(z). By proper adjustment of these variables it is possible to make the weightihg more effective. This is accomplished by making γ_1 and γ_2 a function of the spectral shape of the input signal. This adaptation is done once per 10 ms frame, but an interpolation procedure for each first subframe is used to smooth this adaptation process. The spectral shape is obtained from a 2nd-order linear prediction filter, obtained as a by product from the Levinson-Durbin recursion (Subsection III.3.2.2). The reflection coefficients k_i, are converted to Log Area Ratio (LAR,) coefficients o_i by

$$o_i=\log\frac{(1.0+k_i)}{(1.0-k_i)}\qquad i=1,2. \tag{28}$$

These LAR, coefficients are used for the second subframe. The LAR, coefficients for the first subframe are obtained through linear interpolation with the LAR, parameters from the previous frame, and are given by:

$$\text{Subframe 1:}\quad o1_i=\quad 0.5o_i^{(m-1)}+0.5o_i^{(m)},\qquad i=1,\ldots,2, \tag{29}$$

$$\text{Subframe 2:}\quad o2_i=\quad o_i^{(m)},\qquad i=1,\ldots,2.$$

The spectral envelope is characterized as being either flat (flat=1) or tilted (flat=0). For each subframe this characterization is obtained by applying a threshold function to the LAR coefficients. To avoid rapid changes, a hysteresis is used by taking into account the value of flat in the previous subframe (m–1),

$$flat^{(m)}=\begin{cases}0 & \text{if } o_1<-1.74 \text{ and } o_2>0.65 \text{ and } flat^{(m-1)}=1,\\ 1 & \text{if } o_1>-1.52 \text{ and } o_2<0.43 \text{ and } flat^{(m-1)}=0,\\ flat^{(m-1)} & \text{otherwise.}\end{cases} \tag{30}$$

If the interpolated spectrum for a subframe is classified as flat ($\text{flat}^{(m)}=1$), the weight factors are set to $\gamma_1=0.94$ and $\gamma_2=0.6$. If the spectrum is classified as tilted ($\text{flat}^{(m)}=0$), the value of γ_1 is set to 0.98, and the value of γ_2 is adapted to the strength of the resonances in the LP synthesis filter, but is bounded between 0.4 and 0.7. If a strong resonance is present, the value of γ_2 is set closer to the upperbound. This adaptation is achieved by a criterion based on the minimum distance between 2 successive LSP coefficients for the current subframe. The minimum distance is given by

$$d_{min}=min[\omega_{i+1}-\omega_i]i=1,\ldots,9. \tag{31}$$

The following linear relation is used to compute γ_2:

$$\gamma_2=6.0^{*}d_{min}+1.0,\ and\ 0.4\leqq\gamma_2\leqq 0.7 \tag{32}$$

The weighted speech signal in a subframe is given by

$$sw(n)=s(n)+\sum_{i=1}^{10}a_i\gamma_1^{i}s(n-i)-\sum_{i=1}^{10}a_i\gamma_2^{i}sw(n-i), \tag{33}$$

$$n=0,\ldots,39.$$

The weighted speech signal sw(n) is used to find an estimation of the pitch delay in the speech frame.

III.3.4 Open-Loop Pitch Analysis

To reduce the complexity of the search for the best adaptive codebook delay, the search range is limited around a candidate delay T_{op}, obtained from an open-loop pitch analysis. This open-loop pitch analysis is done once per frame (10 ms). The open-loop pitch estimation uses the weighted speech signal sw(n) of Eq. (33), and is done as follows: In the first step, 3 maxima of the correlation

$$R(k)=\sum_{n=0}^{79}sw(n)sw(n-k) \tag{34}$$

are found in the following three ranges

$$i=1:80,\ldots,143,$$
$$i=2:40,\ldots,79,$$
$$i=3:20,\ldots,39.$$

The retained maxima $R(t_i)$, i=1, . . . , 3, are normalized through

$$R'(t_i)=\frac{R(t_i)}{\sqrt{\sum_n sw^2(n-t_i)}},\qquad i=1,\ldots,3, \tag{35}$$

The winner among the three normalized correlations is selected by favoring the delays with the values in the lower range. This is done by weighting the normalized correlations corresponding to the longer delays. The best open-loop delay T_{op} is determined as follows:

```
$T_{op}=t_1$
$R'(T_{op})=R'(t_1)$
if $R'(t_2)\geqq 0.85R'(T_{op})$
    $R'(T_{op})=R'(t_2)$
     $T_{op}=t_2$
end
if $R'(t_3)\geqq 0.85R(T_{op})$
    $R'(T_{op})=R'(t_3)$
     $T_{op}=t_3$
end
```

This procedure of dividing the delay range into 3 sections and favoring the lower sections is used to avoid choosing pitch multiples.

III.3.5 Computation of the Impulse Response

The impulse response, h(n), of the weighted synthesis filter $W(z)/\hat{A}(z)$ is computed for each subframe. This impulse response is needed for the search of adaptive and fixed codebooks. The impulse response h(n) is computed by filtering the vector of coefficients of the filter $A(z/\gamma_1)$ extended by zeros through the two filters $1/\hat{A}(z)$ and $1/A(z/\gamma_2)$.

III.3.6Computation of the Target Signal

The target signal x(n) for the adaptive codebook search is usually computed by subtracting the zero-input response of the weighted synthesis filter $W(z)/\hat{A}(z)=A(z/\gamma_1)/[\hat{A}(z)A(z/\gamma_2)]$ from the weighted speech signal sw(n) of Eq. (33). This is done on a subframe basis.

An equivalent procedure for computing the target signal, which is used in this Recommendation, is the filtering of the LP residual signal r(n) through the combination of synthesis filter $1/\hat{A}(z)$ and the weighting filter $A(z/\gamma_1)/A(z/\gamma_2)$. After determining the excitation for the subframe, the initial states of these filters are updated by filtering the difference between the LP residual and excitation. The memory update of these filters is explained in Subsection III.3.10.

The residual signal r(n), which is needed for finding the target vector is also used in the adaptive codebook search to extend the past excitation buffer. This simplifies the adaptive codebook search procedure for delays less than the subframe size of 40 as will be explained in the next section. The LP residual is given by

$$r(n)=s(n)+\sum_{i=1}^{10}\hat{a}_i s(n-i), \qquad n=0,\ldots,39. \tag{36}$$

III.3.7 Adaptive-Codebook Search

The adaptive-codebook parameters (or pitch parameters) are the delay and gain. In the adaptive codebook approach for implementing the pitch filter, the excitation is repeated for delays less than the subframe length. In the search stage, the excitation is extended by the LP residual to simplify the closed-loop search. The adaptive-codebook search is done every (5 ms) subframe. In the first subframe, a fractional pitch delay T_1 is used with a resolution of ⅓ in the range [19⅓, 84⅔] and integers only in the range [85, 143]. For the second subframe, a delay T_2 with a resolution of ⅓ is always used in the range [(int)T_1−5⅔, (int)T_1+4⅔], where (int)T_1 is the nearest integer to the fractional pitch delay T_1 of the first subframe. This range is adapted for the cases where T_1 straddles the boundaries of the delay range.

For each subframe the optimal delay is determined using close&loop analysis that minimizes the weighted mean-squared error. In the first subframe the delay T_1 is found be searching a small range (6 samples) of delay values around the open-loop delay T_{op} (see Subsection III.3.7). The search boundaries t_{min} and t_{max} are defined by

```
$t_{min}=T_{op}-3$

if $t_{min} < 20$ then $t_{min} = 20$
$t_{max} = t_{min} + 6$
if $t_{max} > 143$ then
        $t_{max} = 143$
        $t_{min} = t_{max} - 6$
end
```

For the second subframe, closed-loop pitch analysis is done around the pitch selected in the first subframe to find the optimal delay T_2. The search boundaries are between t_{min}−⅔ and t_{max}+⅔, where t_{min} and t_{max} are derived from T_1 as follows:

```
$t_{min} = (int)T_1 - 5$
if $t_{min} < 20$ then $t_{min} = 20$
$t_{max} = t_{min} + 9$
if $t_{max} > 143$ then
        $t_{max} = 143$
        $t_{min} = t_{max} - 9$
end
```

The closed-loop pitch search minimizes the mean-squared weighted error between the original and synthesized speech. This is achieved by maximizing the term

$$R(k)=\frac{\sum_{n=0}^{39}x(n)y_k(n)}{\sqrt{\sum_{n=0}^{39}y_k(n)y_k(n)}}, \tag{37}$$

where x(n) is the target signal and $y_k(n)$ is the past filtered excitation ae delay k (past excitation convolved with h(n)). Note that the search range is limited around a preselected value, which is the open-loop pitch T_{op} for the first subframe, and T_1 for the second subframe.

The convolution $y_k(n)$ is computed for the delay t_{min}, and for the other integer delays in the search range k=t_{min}+1, . . . , t_{max}, it is updated using the recursive relation

$$y_k(n)=y_{k-1}(n-1)+u(-k)h(n),\ n=39,\ldots,0, \tag{38}$$

where u(n), n=−143, . . . ,39, is the excitation buffer, and $y_{k-1}(-1)=0$. Note that in the search stage, the samples u(n), n=0, . . . ,39 are not known, and they are needed for pitch delays less than 40. To simplify the search, the LP residual is copied to u(n) to make the relation in Eq. (38) valid for all delays.

For the determination of T_2, and T_1 if the optimum integer closed-loop delay is less than 84, the fractions around the optimum integer delay have to be tested. The fractional pitch search is done by interpolating the normalized correlation in Eq. (37) and searching for its maximum. The interpolation is done using a FIR filter b_{12} based on a Hamming windowed sine function with the sine truncated at ±11 and padded with zeros at ±12 ($b_{12}(12)=0$). The filter has its cut-off frequency (−3 dB) at 3600 Hz in the oversampled domain. The interpolated values of R(k) for the fractions −⅔, −⅓, 0, ⅓, and ⅔ are obtained using the interpolation formula

$$R(k)_t=\sum_{i=0}^{3}R(k-i)b_{12}(t+i.3)+\sum_{i=0}^{3}R(k+1+i)b_{12}(3-t+i.3), \tag{39}$$

$$t=0,\,1,\,2,$$

where t=0, 1, 2 corresponds to the fractions 0, ⅓, and ⅔, respectively. Note that it is necessary to compute correlation terms in Eq. (37) using a range t_{min}−4,t_{max}+4, to allow for the proper interpolation.

III.3.7.1 Generation of the Adaptive Codebook Vector

Once the noninteger pitch delay has been determined, the adaptive codebook vector v(n) is computed by interpolating the past excitation signal u(n) at the given integer delay k and fraction t

$$v(n)=\sum_{i=0}^{9}u(n-k+i)b_{30}(t+i.3)+\sum_{i=0}^{9}u(n-k+1+i)b_{30}(3-t+i.3), \tag{40}$$

$$n=0,\,1,\,2.$$

The interpolation filter b_{30} is based on a Hamming windowed sine functions with the sine truncated at ±29 and padded with zeros at ±30 ($b_{30}(30)=0$). The filters has a cut-off frequency (−3 dB) at 3600 Hz in the oversampled domain.

III.3.7.2 Codeword Computation for Adaptive Codebook Delays

The pitch delay T_1 is encoded with 8 bits in the first subframe and the relative delay in the second subframe is encoded with 5 bits. A fractional delay T is represented by its integer part (int)T, and a fractional part frac/3, frac=−1, 0, 1. The pitch index P1 is now encoded as

$$P1=\begin{cases}((int)T_1-19)*3+frac-1, & \\ \qquad \text{if } T_1=[19,\ldots,85], frac=[-1,0,1] & \\ ((int)T_1-85)+197, & \\ \qquad \mathit{if}\ T_1=[86,\ldots,143], frac=0 & \end{cases} \tag{41}$$

The value of the pitch delay T_2 is encoded relative to the value of T_1. Using the same interpretation as before, the fractional delay T_2 represented by its integer part (int)T_2, and a fractional part frac/3, frac=−1, 0, 1, is encoded as

$$P2=((int)T_2-t_{min})*3+frac+2 \tag{42}$$

where t_{min} is derived from T_1 as before.

To make the coder more robust against random bit errors, a parity bit P0 is computed on the delay index of the first subframe. The parity bit is generated through an XOR operation on the 6 most significant bits of P1. At the decoder this parity bit is recomputed and if the recomputed value does not agree with the transmitted value, an error concealment procedure is applied.

III.3.7.3 Computation of the Adaptive-Codebook Gain

Once the adaptive-codebook delay is determined, the adaptive-codebook gain g_p is computed as

$$g_p=\frac{\sum_{n=0}^{39}x(n)y(n)}{\sum_{n=0}^{39}y(n)y(n)},\quad\text{bounded by } 0\leqq g_p\leqq 1.2, \tag{43}$$

where y(n) is the filtered adaptive codebook vector (zero-state response of $W(z)/\hat{A}(z)$ to v(n)). This vector is obtained by convolving v(n) with h(n)

$$y(n)=\sum_{i=0}^{n}v(i)h(n-i)\quad n=0,\ldots,39. \tag{44}$$

Note that by maximizing the term in Eq. (37) in most cases $g_p>0$: In case the signal contains only negative correlations, the value of g_p is set to 0.

III.3.8 Fixed Codebook: Structure and Search

The fixed codebook is based on an algebraic codebook structure using an interleaved single-pulse permutation (ISPP) design. In this codebook, each codebook vector contains 4 non-zero pulses. Each pulse can have either the amplitudes +1 or −1, and can assume the positions given in Table 7.

The codebook vector c(n) is constructed by taking a zero vector, and putting the 4 unit pulses at the found locations, multiplied with their corresponding sign.

$$c(n)=s0\delta(n-i0)+s1\delta(n-i1)+s2\delta(n-i2)+s3\delta(n-i3),\ n=0,\ldots,39. \tag{45}$$

where δ(0) is a unit pulse. A special feature incorporated in the codebook is that the selected codebook vector is filtered through an adaptive pre-filter P(z) which enhances harmonic components to improve the synthesized speech quality. Here the filter

$$P(z)=1/(1-\beta z^{-T}) \tag{46}$$

TABLE 7

Structure of fixed codebook C.

Pulse	Sign	Positions
10	s0	0, 5, 10, 15, 20, 25, 30, 35
11	s1	1, 6, 11, 16, 21, 26, 31, 36
12	s2	2, 7, 12, 17, 22, 27, 32, 37
13	s3	3, 8, 13, 18, 23, 28, 33, 38
		4, 9, 14, 19, 24, 29, 34, 39

is used, where T is the integer component of the pitch delay of the current subframe, and β is a pitch gain. The value of β is made adaptive by using the quantized adaptive codebook gain from the previous subframe bounded by 0.2 and 0.8.

$$\beta=g_p^{(m-1)},\ 0.2\leqq\beta\leqq 0.8. \tag{47}$$

This filter enhances the harmonic structure for delays less than the subframe size of 40. This modification is incorporated in the fixed codebook search by modifying the impulse response h(n), according to

$$h(n)=h(n)+\beta h(n-T),\ n=T,\ldots,39. \tag{48}$$

III.3.8.1 Fixed-Codebook Search Procedure

The fixed codebook is searched by minimizing the mean-squared error between the weighted input speech sw(n) of Eq. (33), and the weighted reconstructed speech. The target signal used in the closed-loop pitch search is updated by subtracting the adaptive codebook contribution. That is

$$x_2(n)=x(n)-g_py(n),\ n=0,\ldots,39, \tag{49}$$

where y(n) is the filtered adaptive codebook vector of Eq. (44).

The matrix H is defined as the lower triangular Toepliz convolution matrix with diagonal h(0) and lower diagonals h(1), . . . , h(39). If c_k is the algebraic codevector at index k, then the codebook is searched by maximizing the term

$$\frac{C_k^2}{E_k}=\frac{\left(\sum_{n=0}^{39}d(n)c_k(n)\right)^2}{c_k{}^t\Phi c_k}, \tag{50}$$

where d(n) is the correlation between the target signal $x_2(n)$ and the impulse response h(n), and $\Phi=H^tH$ is the matrix of correlations of h(n). The signal d(n) and the matrix Φ are computed before the codebook search. The elements of d(n) are computed from

$$d(n)=\sum_{i=n}^{39}x(i)h(i-n),\quad n=0,\ldots,39, \tag{51}$$

and the elements of the symmetric matrix Φ are computed by

$$\phi(i,j)=\sum_{n=j}^{39}h(n-i)h(n-j),\quad (j\geqq i). \tag{52}$$

Note that only the elements actually needed are computed and an efficient storage procedure has been designed to speed up the search procedure.

The algebraic structure of the codebook C allows for a fast search procedure since the codebook vector c_k contains only four nonzero pulses. The correlation in the numerator of Eq. (50) for a given vector c_k is given by

$$C=\sum_{i=0}^{3}a_id(m_i), \tag{53}$$

where m_i is the position of the ith pulse and a_i is its amplitude. The energy in the denominator of Eq. (50) is given by

$$E=\sum_{i=0}^{3}\phi(m_i,m_i)+2\sum_{i=0}^{2}\sum_{j=i+1}^{3}a_ia_j\phi(m_i,m_j). \tag{54}$$

To simplify the search procedure, the pulse amplitudes are predetermined by quantizing the signal d(n). This is done by setting the amplitude of a pulse at a certain position equal to the sign of d(n) at that position. Before the codebook search, the following steps are done. First, the signal d(n) is decomposed into two signals: the absolute signal d'(n)=|d(n)| and the sign signal sign[d(n)]. Second, the matrix Φ is modified by including the sign information; that is,

$$\phi'(i,j)=\text{sign}[d(i)]\text{sign}[d(j)]\phi(i,j),\ i=0,\ldots,39,\ j=i,\ldots,39. \tag{55}$$

To remove the factor 2 in Eq. (54)

$$\phi'(i,i)=0.5\phi(i,i),\ i=0, \ldots ,39. \tag{56}$$

The correlation in Eq. (53) is now given by

$$C=d'(m_0)+d'(m_1)+d'(m_2)+d'(m_3), \tag{57}$$

and the energy in Eq. (54) is given by

$$\begin{aligned} E=&\phi'(m_0,m_0)\\ &+\phi'(m_1,m_1)+\phi'(m_0,m_1)\\ &+\phi'(m_2,m_2)+\phi'(m_0,m_2)+\phi'(m_1m_2)\\ &+\phi'(m_3,m_3)+\phi'(m_1,m_3)+\phi'(m_2,m_3). \end{aligned} \tag{58}$$

A focused search approach is used to further simplify the search procedure. In this approach a procomputed threshold is tested before entering the last loop, and the loop is entered only if this threshold is exceeded. The maximum number of times the loop can be entered is fixed so that a low percentage of the codebook is searched. The threshold is computed based on the correlation C. The maximum absolute correlation and the average correlation due to the contribution of the first three pulses, max_3 and av_3, are found before the codebook search. The threshold is given by

$$thr_3=av_3+K_3(max_3-av_3). \tag{59}$$

The fourth loop is entered only if the absolute correlation (due to three pulses) exceeds thr_3, where $0 \leqq K_3 < 1$. The value of K_3 controls the percentage of codebook search and it is set here to 0.4. Note that this results in a variable search time, and to further control the search the number of times the last loop is entered (for the 2 subframes) cannot exceed a certain maximum, which is set here to 180 (the average worst case per subframe is 90 times).

III.3.8.2 Codeword Computation of the Fixed Codebook

The pulse positions of the pulses i0, i1, and i2, are encoded with 3 bits each, while the position of i3 is encoded with 4 bits. Each pulse amplitude is encoded with 1 bit. This gives a total of 17 bits for the 4 pulses. By defining s=1 if the sign is positive and s=0 is the sign is negative, the sign codeword is obtained from

$$S=s0+2*s1+4*s2+8*s3 \tag{60}$$

and the fixed codebook codeword is obtained from

$$C=(i0/5)+8*(i1/5)+64*(i2/5)+512*(2*(i3/5)+jx) \tag{61}$$

where jx=0 if i3=3,8, . . . , and jz=1 if i3=4,9, . . .

III.3.9 Quantization of the Gains

The adaptive-codebook gain (pitch gain) and the fixed (algebraic) codebook gain are vector quantized using 7 bits. The gain codebook search is done by minimizing the mean-squared weighted error between original and reconstructed speech which is given by

$$E=x^tx+g_p^{\,2}Y^ty+g_c^{\,2}z^tz-2g_px^ty-2g_cx^tz+2g_pg_cy^tz+tm \tag{62}$$

where x is the target vector (see Subsection III.3.6), y is the filtered adaptive codebook vector of Eq. (44), and z is the fixed codebook vector convolved with h(n),

$$z(n)=\sum_{i=0}^{n} c(i)h(n-i) \quad n=0, \ldots, 39. \tag{63}$$

III.3.9.1 Gain Prediction

The fixed codebook gain gc can be expressed as

$$g_c=\gamma g'_c, \tag{64}$$

where g'_c is a predicted gain based on previous fixed codebook energies, and γ is a correction factor.

The mean energy of the fixed codebook contribution is given by

$$E=10\log\left(\frac{1}{40}\sum_{i=0}^{39} c_i^2\right). \tag{65}$$

After scaling the vector c_i with the fixed codebook gain g_c, the energy of the scaled fixed codebook is given by 20 log g_c+E. Let $E^{(m)}$ be the mean-removed energy (in dB) of the (scaled) fixed codebook contribution at subframe m, given by

$$E^{(m)}=20\log g_c+E-\bar{E}, \tag{66}$$

where $\bar{E}$=30 dB is the mean energy of the fixed codebook excitation. The gain g_c can be expressed as a function of $E^{(m)}$, E, and $\bar{E}$ by

$$g_c=10^{(E^{(m)}+\bar{E}-E)/20} \tag{67}$$

The predicted gain g'_c is found by predicting the log-energy of the current fixed codebook contribution from the log-energy of previous fixed codebook contributions. The 4th order MA prediction is done as follows. The predicted energy is given by

$$\tilde{E}^{(m)}=\sum_{i=1}^{4} b_i\hat{R}^{(m-i)}, \tag{68}$$

where $[b_1\ b_2\ b_3\ b_4]$=[0.68 0.58 .034 0.19] are the MA prediction coefficients, and $\hat{R}^{(m)}$ is the quantized version of the prediction error $R^{(m)}$ at subframe m, defined by

$$R^{(m)}=E^{(m)}-\tilde{E}^{(m)}. \tag{69}$$

The predicted gain g'_c is found by replacing $E^{(m)}$ by its predicted value in Eq (67).

$$g'_c=10^{\tilde{E}^{(m)}+\bar{E}-E/20}. \tag{70}$$

The correction factor γ is related to the gain-prediction error by

$$R^{(m)}=E^{(m)}-\tilde{E}^{(m)}=20\log(\gamma). \tag{71}$$

III.3.9.2 Codebook Search for Gain Quantization

The adaptive-codebook gain, g_p, and the factor γ are vector quantized using a 2-stage conjugate structured codebook. The first stage consists of a 3 bit two-dimensional codebook GA, and the second stage consists of a 4 bit two-dimensional codebook GB. The first element in each codebook represents the quantized adaptive codebook gain $\hat{g}_p$, and the second element represents the quantized fixed codebook gain correction factor $\hat{\gamma}$. Given codebook indices m and n for GA and GB, respectively, the quantized adaptive-codebook gain is given by

$$\hat{g}_p=GA_1(m)+GB_1(n), \tag{72}$$

and the quantized fixed-codebook gain by

$$\hat{g}_c=g'_c\hat{\gamma}=g'_c(GA_2(m)+GB_2(n)). \qquad (73)$$

This conjugate structure simplifies the codebook search, by applying a pre-selection process. The optimum pitch gain g_p, and fixed-codebook gain, g_c, are derived from Eq. (62), and are used for the pre-selection. The codebook GA contains 8 entries in which the second element (corresponding to g_c) has in general larger values than the first element (corresponding to g_p). This bias allows a pre-selection using the value of g_c. In this pre-selection process, a cluster of 4 vectors whose second element are close to gx_c, where gx_c is derived from g_c and g_p. Similarly, the codebook GB contains 16 entries in which have a bias towards the first element (corresponding to g_p). A cluster of 8 vectors whose first elements are close to g_p are selected. Hence for each codebook the best 50% candidate vectors are selected. This is followed by an exhaustive search over the remaining 4*8=32 possibilities, such that the combination of the two indices minimizes the weighted mean-squared error of Eq. (62).

III.3.9.3 Codeword Computation for Gain Quantizer

The codewords GA and GB for the gain quantizer are obtained from the indices corresponding to the best choice. To reduce the impact of single bit errors the codebook indices are mapped.

III.3.10 Memory Update

An update of the states of the synthesis and weighting filters is needed to compute the target signal in the next subframe. After the two gains are quantized, the excitation signal, u(n), in the present subframe is found by

$$u(n)=\hat{g}_p v(n)+\bar{g}_c c(n),\ n=0, \ldots, 39, \qquad (74)$$

where $\hat{g}_p$ and $\hat{g}_c$ are the quantized adaptive and fixed codebook gains, respectively, v(n) the adaptive codebook vector (interpolated past excitation), and c(n) is the fixed codebook vector (algebraic codevector including pitch sharpening). The states of the filters can be updated by filtering the signal r(n)−u(n) (difference between residual and excitation) through the filters $1/\hat{A}(z)$ and $A(z/\gamma_1)/A(z/\gamma_2)$ for the 40 sample subframe and saving the states of the filters. This would require 3 filter operations. A simpler approach, which requires only one filtering is as follows. The local synthesis speech, $\hat{s}(n)$, is computed by filtering the excitation signal through $1/\hat{A}(z)$. The output of the filter due to the input r(n)−u(n) is equivalent to $e(n)=s(n)-\hat{s}(n)$. So the states of the synthesis filter $1/\hat{A}(z)$ are given by e(n), n=30, . . . ,39. Updating the states of the filter $A(z/\gamma_1)/A(z/\gamma_2)$ can be done by filtering the error signal e(n) through this filter to find the perceptually weighted error ew(n). However, the signal ew(n) can be equivalently found by

$$ew(n)=x(n)-\hat{g}_p y(n)+\hat{g}_c z(n) \qquad (75)$$

Since the signals z(n), y(n), and z(n) are available, the states of the weighting filter are updated by computing ew(n) as in Eq. (75) for n=30, . . . ,39. This saves two filter operations.

III.3.11 Encoder and Decoder Initialization

All static encoder variables should be initialized to 0, except the variables listed in table 8. These variables need to be initialized for the decoder as well.

TABLE 8

Description of parameters with nonzero initialization

Variable	Reference	Initial value
β	Section 3.8	0.8
l_i	Section 3.2.4	iπ/11
q_i	Section 3.2.4	0.9595, . . . ,
$\hat{R}^{(k)}$	Section 3.9.1	−14

III.4.0 FUNCTIONAL DESCRIPTION OF THE DECODER

The signal flow at the decoder was shown in Subsection III.2 (FIG. 4). First the parameters are decoded (LP coefficients, adaptive codebook vector, fixed codebook vector, and gains). These decoded parameters are used to compute the reconstructed speech signal. This process is described in Subsection III.4.1. This reconstructed signal is enhanced by a post-processing operation consisting of a postfilter and a high-pass filter (Subsection III.4.2). Subsection III.4.3 describes the error concealment procedure used when either a parity error has occurred, or when the frame erasure flag has been set.

III.4.1 Parameter Decoding Procedure

The transmitted parameters are listed in Table 9. At startup all static encoder variables should be initialized to 0, except the variables listed in Table 8. The decoding process is done in the following order:

TABLE 9

Description of transmitted parameters indices. The bitstream ordering is reflected by the order in the table. For each parameter the most significant bit (MSB) is transmitted first.

Symbol	Description	Bits
L0	Switched predictor index of LSP quantizer	1
L1	First stage vector of LSP quantizer	7
L2	Second stage lower vector of LSP quantizer	5
L3	Second stage higher vector of LSP quantizer	5
P1	Pitch delay 1st subframe	8
P0	Parity bit for pitch	1
S1	Signs of pulses 1st subframe	4
C1	Fixed codebook 1st subframe	13
GA1	Gain codebook (stage 1) 1st subframe	3
GB1	Gain codebook (stage 2) 1st subframe	4
P2	Pitch delay 2nd subframe	5
S2	Signs of pulses 2nd subframe	4
C2	Fixed codebook 2nd subframe	13
GA2	Gain codebook (stage 1) 2nd subframe	3
GB2	Gain codebook (stage 2) 2nd subframe	4

III.4.1.1 Decoding of LP Filter Parameters

The received indices **L0**, **L1**, **L2**, and **L3** of the LSP quantizer are used to reconstruct the quantized LSP coefficients using the procedure described in Subsection III.3.2.4. The interpolation procedure described in Subsection III.3.2.5 is used to obtain 2 interpolated LSP vectors (corresponding to 2 subframes). For each subframe, the interpolated LSP vector is converted to LP filter coefficients a_i, which are used for synthesizing the reconstructed speech in the subframe.

The following steps are repeated for each subframe:

1. decoding of the adaptive codebook vector,

2. decoding of the fixed codebook vector,

3. decoding of the adaptive and fixed codebook gains,
4. computation of the reconstructed speech,

III.4.1.2 Decoding of the Adaptive Codebook Vector

The received adaptive codebook index is used to find the integer and fractional parts of the pitch delay. The integer part (int)T_1 and fractional part frac of T_1 are obtained from P1 as follows:

```
if $P1 < 197$
    $(int)T_1 = (P1 + 2)/3 + 19$
    $frac = P1 - (int)T_1 * 3 + 58$
else
    $(int)T_1 = P1 - 112$
    $frac = 0$
end
```

The integer and fractional part of T_2 are obtained from P2 and t_{min}, where t_{min} is derived from P1 as follows

```
$t_{min} = (int)T_1 - 5$
if $t_{min} < 20$ then $t_{min} = 20$
$t_{max} = t_{min} + 9$
if $t_{max} > 143$ then
    $t_{max} = 143$
    $t_{min} = t_{max} - 9$
end
```

Now T2 is obtained from

$$(int)T_2=(P2+2)/3-1+t_{min}$$

$$frac=P2-2-((P2+2)/3-1)*3$$

The adaptive codebook vector v(n) is fonnd by interpolating the past excitation u(n) (at the pitch delay) using Eq. (40).

III.4.1.3 Decoding of the Fixed Codebook Vector

The received fixed codebook index C is used to extract the positions of the excitation pulses. The pulse signs are obtained from S. Once the pulse positions and signs are decoded the fixed codebook vector c(n), can be constructed. If the integer part of the pitch delay, T, is less than the subframe size 40, the pitch enhancement procedure is applied which modifies c(n) according to Eq. (48).

III.4.1.4 Decoding of the Adaptive and Fixed Codebook Gains

The received gain codebook index gives the adaptive codebook gain $\hat{g}_p$ and the fixed codebook gain correction factor $\hat{\gamma}$. This procedure is described in detail in Subsection III.3.9. The estimated fixed codebook gain g'_c is found using Eq. (70). The fixed codebook vector is obtained from the product of the quantized gain correction factor with this predicted gain (Eq. (64)). The adaptive codebook gain is reconstructed using Eq. (72).

III.4.1.5 Computation of the Parity Bit

Before the speech is reconstructed, the parity bit is recomputed from the adaptive codebook delay (Subsection III.3.7.2). If this bit is not identical to the transmitted parity bit P0, it is likely that bit errors occurred during transmission and the error concealment procedure of Subsection III.4.3 is used.

III.4.1.6 Computing the Reconstructed Speech

The excitation u(n) at the input of the synthesis filter (see Eq. (74)) is input to the LP synthesis filter. The reconstructed

speech for the subframe is given by

$$\hat{s}(n) = u(n) - \sum_{i=1}^{10} \hat{a}_i \hat{s}(n-i), \quad n = 0, \ldots, 39. \tag{76}$$

where $\hat{a}_i$ are the interpolated LP filter coefficients.

The reconstructed speech $\hat{s}(n)$ is then processed by a post processor which is described in the next section.

III.4.2 Post-Processing

Post-processing consists of three functions: adaptive postfiltering, high-pass filtering, and signal up-scaling. The adaptive postfilter is the cascade of three filters: a pitch postfilter $H_p(z)$, a short-term postfilter $H_f(z)$, and a tilt compensation filter $H_t(z)$, followed by an adaptive gain control procedure. The postfilter is updated every subframe of 5 ms. The postfiltering process is organized as follows. First, the synthesis speech $\hat{s}(n)$ is inverse filtered through $\hat{A}(z/\gamma_n)$ to produce the residual signal $\hat{r}(n)$. The signal $\hat{r}(n)$ is used to compute the pitch delay T and gain g_{pit}. The signal $\hat{r}(n)$ is filtered through the pitch postfilter $H_p(z)$ to produce the signal r'(n) which, in its turn, is filtered by the synthesis filter $1/[g_f\hat{A}(z/\gamma_d)]$. Finally, the signal at the output of the synthesis filter $1/[g_f\hat{A}(z/\gamma_d)]$ is passed to the tilt compensation filter $H_t(z)$ resulting in the postfiltered synthesis speech signal sf(n). Adaptive gain controle is then applied between sf(n) and $\hat{s}(n)$ resulting in the signal sf'(n). The high-pass filtering and scaling operation operate on the post filtered signal sf'(n) .

III.4.2.1 Pitch Postfilter

The pitch, or harmonic, postfilter is given by

$$H_p(z) = \frac{1}{1+g_0}\,(1+g_0z^{-T}), \tag{77}$$

where T is the pitch delay and go is a gain factor given by

$$g_0=\gamma_p g_{pit} \tag{78}$$

where g_{pit} is the pitch gain. Both the pitch delay and gain are determined from the decoder output signal. Note that g_{pit} is bounded by 1, and it is set to zero if the pitch prediction gain is less that 3 dB. The factor γ_p controls the amount of harmonic postfiltering and has the value γ_p=0.5. The pitch delay and gain are computed from the residual signal $\hat{r}(n)$ obtained by filtering the speech $\hat{s}(n)$ through $\hat{A}(z/\gamma_n)$, which is the numerator of the short-term postfilter (see Subsection III.4.2.2)

$$\hat{r}(n) = \hat{s}(n) + \sum_{i=1}^{10} \gamma_n{}^i \hat{a}_i \hat{s}(n-i). \tag{79}$$

The pitch delay is computed using a two pass procedure. The first pass selects the best integer in the range $[T_1-1,T_1+1]$, where T_1 is the integer part of the (transmitted) pitch delay in the first subframe. The best integer delay is the one that maximizes the correlation

$$R(k) = \sum_{n=0}^{39} \hat{r}(n)\hat{r}(n-k). \tag{80}$$

The second pass chooses the best fractional delay T with resolution ⅛ around T_0. This is done by finding the delay with the highest normalized correlation.

$$R'(k)=\frac{\sum_{n=0}^{39}\hat{r}(n)\hat{r}_k(n)}{\sqrt{\sum_{n=0}^{39}\hat{r}_k(n)\hat{r}_k(n)}}, \tag{81}$$

where r̂(n) is the residual signal at delay k. Once the optimal delay T is found, the corresponding correlation value is compared against a threshold. If R'(T)<0.5 then the harmonic postfilter is disabled by setting g_{plt}=0. Otherwise the value of g_{pit} is computed from:

$$g_{plt}=\frac{\sum_{n=0}^{39}\hat{r}(n)\hat{r}_k(n)}{\sum_{n=0}^{39}\hat{r}_k(n)\hat{r}_k(n)}, \text{ bounded by } 0 \leqq g_{plt} \leqq 1.0. \tag{82}$$

The noninteger delayed signal $\hat{r}_k$(n) is first computed using an interpolation filter d length 33. After the selection of T, $\hat{r}_k$(n) is recomputed with a longer interpolation filter of length 129. The new signal replaces the previous one only if the longer filter incre.es the value of R'(T).

III.4.2.2 Short-Term Postfilter

The short-term postfilter is given by

$$H_f(z)=\frac{1}{g_f}\ \frac{\hat{A}(z/\gamma_n)}{\hat{A}(z/\gamma_d)}=\frac{1}{g_f}\ \frac{1+\sum_{i=1}^{10}\gamma_n{}^i\hat{a}_i z^{-i}}{1+\sum_{i=1}^{10}\gamma_d{}^i\hat{a}_i z^{-i}}, \tag{83}$$

where Â(z) is the received quantized LP inverse filter (LP analysis is not done at the decoder), and the factors γ_n and γ_d control the amount of short-term postfiltering, and are set to γ_n=0.55, and γ_d=0.7. The gain term g_f is calculated on the truncated impulse response, h_f(n), d the filter Â(z/γ_n)/Â(z/γ_d) are given by

$$g_f=\sum_{n=0}^{19}|h_f(n)|. \tag{84}$$

III.4.2.3 Tilt Compensation

Finally, the filter H_t(z) compensates for the tilt in the short-term postfilter H_f(z) and is given by

$$H_t(z)=\frac{1}{g_t}\ (1+\gamma_t k_1 z^{-1}), \tag{85}$$

where $\gamma_t k_1$ is a tilt factor, k_1 being the first reflection coefficient calculated on h_f(n) with

$$k_1=-\frac{r_h(1)}{r_h(0)}\ ;\ r_h(i)=\sum_{j=0}^{19-i}h_f(j)h_f(j+i). \tag{86}$$

The gain term g_t=1−|$\gamma_t k_1$| compensates for the decreasing effect of g_f in H_f(z). Furthermore, it has been shown that the product filter H_f(z)H_t(z) has generally no gain.

Two values for γ_t are used depending on the sign of k_1. If k_1 is negative, γ_t=0.9, and if k_1 is positive, γ_t=0.2.

III.4.2.4 Adaptive Gain Control

Adaptive gain control is used to compensate for gain differences between the reconstructed speech signal ŝ(n) and the postfiltered signal sf(n). The gain scaling factor G for the present subframe is computed by

$$G=\frac{\sum_{n=0}^{39}|\hat{s}(n)|}{\sum_{n=0}^{39}|sf(n)|}. \tag{87}$$

The gain-scaled postfiltered signal sf'(n) is given by

$$sf'(n)=g(n)sf(n),\ n=0,\ \ldots,39, \tag{88}$$

where g(n) is updated on a sample-by-sample basis and given by

$$g(n)=0.85g(n-1)+0.15G,\ n=0,\ \ldots,39. \tag{89}$$

The initial value of g(−1)=1.0.

III.4.2.5 High-pass Filtering and Up-Scaling

A high-pass filter at a cutoff frequency of 100 Hz is applied to the reconstructed and postfiltered speech sf'(n). The filter is given by

$$H_{h2}(z)=\frac{0.93980581-1.8795834z^{-1}+0.93980581z^{-2}}{1-1.9330735z^{-1}+0.93589199z^{-2}}. \tag{90}$$

Up-scaling consists of multiplying the high-pass filtered output by a factor 2 to retrieve the input signal level.

III.4.3 Concealment of Frame Erasures and Parity Errors

An error concealment procedure has been incorporated in the decoder to reduce the degradations in the reconstructed speech because of frame erasures or random errors in the bitstream. This error concealment process is functional when either i) the frame of coder parameters (corresponding to a 10 ms frame) has been identified as being erased, or ii) a checksum error occurs on the parity bit for the pitch delay index P1. The latter could occur when the bitstream has been corrupted by random bit errors.

If a parity error occurs on P1, the delay value T_1 is set to the value of the delay of the previous frame. The value of T_2 is derived with the procedure outlined in Subsection III.4.1.2, using this new value of T_1. If consecutive parity errors occur, the previous value of T_1, incremented by 1, is used.

The mechanism for detecting frame erasures is not defined in the Recommendation, and will depend on the application. The concealment strategy has to reconstruct the current frame, based on previously received information. The method used replaces the missing excitation signal with one of similar characteristics, while gradually decaying its energy. This is done by using a voicing classifier based on the long-term prediction gain, which is computed as part of the long-term postfilter analysis. The pitch postfilter (see Subsection III.4.2.1) finds the long-term predictor for which the prediction gain is more than 3 dB. This is done by setting a threshold of 0.5 on the normalized correlation R'(k) (Eq. (81)). For the error concealment process, these frames will be classified as periodic. Otherwise the frame is declared nonperiodic. An erased frame inherits its class from the preceding (reconstructed) speech frame. Note that the voicing classification is continuously updated based on this reconstructed speech signal. Hence, for many consecutive erased frames the classification might change. Typically, this only happens if the original classification was periodic.

The specific steps taken for an erased frame are:

1. repetition of the LP filter parameters,

2. attenuation of adaptive and fixed codebook gains,
3. attenuation of the memory of the gain predictor,
4. generation of the replacement excitation.

III.4.3.1 Repetition of LP Filter Parameters

The LP parameters of the last good frame are used. The states of the LSF predictor contain the values of the received codewords l_i. Since the current codeword is not available it is computed from the repeated LSF parameters $\hat{\omega}_i$ and the predictor memory from

$$l_i=\left[\hat{\omega}_i^{(m)}-\sum_{k=1}^{4} m_i^k l_i^{(m-k)}\right] / \left(1-\sum_{k=1}^{4} m_i^k\right), i=1, \ldots, 10. \tag{91}$$

III.4.3.2 Attenuation of Adaptive and Fixed Codebook Gains

An attenuated version of the previous fixed codebook gain is used.

$$g_c^{(m)}=0.98 g_c^{(m-1)}. \tag{92}$$

The same is done for the adaptive codebook gain. In addition a clipping operation is used to keep its value below 0.9.

$$g_p^{(m)}=0.9 g_p^{(m-1)} \text{ and } g_p^{(m)}<0.9. \tag{93}$$

III.4.3.3 Attenuation of the Memory of the Gain Predictor

The gain predictor uses the energy of previously selected codebooks. To allow for a smooth continuation of the coder once good frames are received, the memory of the gain predictor is updated with an attenuated version of the codebook energy. The value of $\hat{R}^{(m)}$ for the current subframe n is set to the averaged quantized gain prediction error, attenuated by 4 dB.

$$\hat{R}^{(m)}=\left(0.25 \sum_{i=1}^{4} \hat{R}^{(m-i)}\right)-4.0 \text{ and } \hat{R}^{(m)} \geqq -14. \tag{94}$$

4.3.4 Generation of the Replacement Excitation

The excitation used depends on the periodicity classification. If the last correctly received frame was classified as periodic, the current frame is considered to be periodic as well. In that case only the adaptive codebook is used, and the fixed codebook contribution is set to zero. The pitch delay is based on the last correctly received pitch delay and is repeated for each successive frame. To avoid excessive periodicity the delay is increased by one for each next subframe but bounded by 143. The adaptive codebook gain is based on an attenuated value according to Eq. (93).

If the last correctly received frame was classified as nonperiodic, the current frame is considered to be nonperiodic as well, and the adaptive codebook contribution is set to zero. The fixed codebook contribution is generated by randomly selecting a codebook index and sign index. The random generator is based on the function

$$seed=seed*31821+13849, \tag{95}$$

with the initial seed value of 21845. The random codebook index is derived from the 13 least significant bits of the next random number. The random sign is derived from the 4 least significant bits of the next random number. The fixed codebook gain is attenuated according to Eq. (92).

III.5 BIT-EXACT DESCRIPTION OF THE CS-ACELP CODER

ANSI C code simulating the CS-ACELP coder in 16 bit fixed-point is available from ITU-T. The following sections summarize the use of this simulation code, and how the software is organized.

III.5.1 Use of the Simulation Software

The C code consists of two main programs coder. c, which simulates the encoder, and decoder. c, which simulates the decoder. The encoder is run as follows:

```
coder inputfile bstreamfile
```

The inputfile and outputfile are sampled data files containing 16-bit PCM signals. The bitstream file contains 81 16-bit words, where the first word can be used to indicate frame erasure, and the remaining 80 words contain one bit each. The decoder takes this bitstream file and produces a post-filtered output file containing a 16-bit PCM signal.

```
decoder bstreamfile outputfile
```

III.5.2 Organization of the Simulation Software

In the fixed-point ANSI C simulation, only two types of fixed-point data are used as is shown in Table 10. To facilitate the implementation of the simulation code, loop indices, Boolean values and flags use the type Flag, which would be either 16 bit or 32 bits depending on the target platform.

TABLE 10

Data types used in ANSI C simulation.

Type	Max. value	Min. value	Description
Word16	0x7fff	0x8000	signed 2's complement 16 bit word
Word32	0x7fffffffL	0x80000000L	signed 2's complement 32 bit word

All the computations are done using a predefined set of basic operators. The description of these operators is given in Table 11. The tables used by the simulation coder are summarized in Table 12. These main programs use a library of routines that are summarized in Tables 13, 14, and 15.

TABLE 11

Basic operations used in ANSI C simulation.

Operation	Description
Word16 sature(Word32 L_var1)	Limit to 16 bits
Word16 add(Word16 var1, Word16 var2)	Short addition
Word16 sub(Word16 var1, Word16 var2)	Short subtraction
Word16 abs_s(Word16 var1)	Short abs

TABLE 11-continued

Basic operations used in ANSI C simulation.

Operation	Description
Word16 shl(Word16 var1, Word16 var2)	Short shift left
Word16 shr(Word16 var1, Word16 var2)	Short shift right
Word16 mult(Word16 var1, Word16 var2)	Short multiplication
Word32 L_mult(Word16 var1, Word16 var2)	Long multiplication
Word16 negate(Word16 var1)	Short negate
Word16 extract_h(Word32 L_var1)	Extract high
Word16 extract_l(Word32 L_var1)	Extract low
Word16 round(Word32 L_var1)	Round
Word32 L_mac(Word32 L_var3, Word16 var1, Word16 var2)	Mac
Word32 L_msu(Word32 L_var3, Word16 var1, Word16 var2)	Msu
Word32 L_macNs(Word32 L_var3, Word16 var1, Word16 var2)	Mac without sat
Word32 L_msuNs(Word32 L_var3, Word16 var1, Word16 var2)	Msu without sat
Word32 L_add(Word32 L_var1, Word32 L_var2)	Long addition
Word32 L_sub(Word32 L_var1, Word32 L_var2)	Long subtraction
Word32 L_add_c(Word32 L_var1, Word32 L_var2)	Long add with c
Word32 L_sub_c(Word32 L_var1, Word32 L_var2)	Long sub with c
Word32 L_negate(Word32 L_var1)	Long negate
Word16 mult_r(Word16 var1, Word16 var2)	Multiplication with round
Word32 L_shl(Word32 L_var1, Word16 var2)	Long shift left
Word32 L_shr(Word32 L_var1, Word16 var2)	Long shift right
Word16 shr_r(Word16 var1, Word16 var2)	Shift right with round
Word16 mac_r(Word32 L_var3, Word16 var1, Word16 var2)	Mac with rounding
Word16 msu_r(Word32 L_var3, Word16 var1, Word16 var2)	Msu with rounding
Word32 L_deposit_h(Word16 var1)	16 bit var1 - MSB
Word32 L_deposit_l(Word16 var1)	16 bit var1 - LSB
Word32 L_shr_r(Word32 L_var1, Word16 var2)	Long shift right with round
Word32 L_abs(Word32L_var1)	Long abs
Word32 L_sat(Word32 L_var1)	Long saturation
Word16 norm_s(Word16 var1)	Short norm
Word16 div_s(Word16 var1, Word16 var2)	Short division
Word16 norm_l(Word32 L_var1)	Long norm

TABLE 12

Summary of tables.

File	Table name	Size	Description
tab_hup.c	tab_hup_s	28	upsampling filter for postfilter
tab_hup.c	tab_hup_l	112	upsampling filter for postfilter
inter_3.c	inter_3	13	FIR filter for interpolating the correlation
pred_lt3.c	inter_3	31	FIR filter for interpolating past excitation
lspcb.tab	lspcb1	128 × 10	LSP quantizer (first stage)
lspcb.tab	lspcb2	32 × 10	LSP quantizer (second stage)
lspcb.tab	fg	2 × 4 × 10	MA predictors in LSP VQ
lspcb.tab	fg_sum	2 × 10	used in LSP VQ
lspcb.tab	fg_sum_inv	2 × 10	used in LSP VQ
qua_gain.tab	gbk1	8 × 2	codebook GA in gain VQ
qua_gain.tab	gbk2	16 × 2	codebook GB in gain VQ
qua_gain.tab	map1	8	used in gain VQ
qua_gain.tab	imap1	8	used in gain VQ
qua_gain.tab	map2	16	used in gain VQ
qua_gain.tab	ima21	16	used in gain VQ
window.tab	window	240	LP analysis window
lag_wind.tab	lag_h	10	lag window for bandwidth expansion (high part)
lag_wind.tab	lag_l	10	lag window for bandwidth expansion (low part)
grid.tab	grid	61	grid points in LP to LSP conversion
inv_sqrt.tab	table	49	lookup table in inverse square root computation
log2.tab	table	33	lookup table in base 2 logarithm computation
lsp_lsf.tab	table	65	lookup table in LSF to LSP conversion and vice versa
lsp_lsf.tab	slope	64	line Slopes in LSP to LSF conversion
pow2.tab	table	33	lookup table in 2 computation
acelp.h			prototypes for fixed codebook search
ld8k.h			prototypes and constants
typedef.h			type definitions

TABLE 13

Summary of encoder specific routines.

Filename	Description
acelp_co.c	Search fixed codebook
autocorr.c	Compute autocorrelation for LP analysis
az_lsp.c	compute LSPs from LP coefficients
cod_ld8k.c	encoder routine
convolve.c	convolution operation
corr_xy2.c	compute correlation terms for gain quantization
enc_lag3.c	encode adaptive codebook index
g_pitch.c	compute adaptive codebook gain
gainpred.c	gain predictor
int_lpc.c	interpolation of LSP
inter_3.c	fractional delay interpolation
lag_wind.c	lag-windowing
levinson.c	levinson recursion
lspenc.c	LSP encoding routine
lspgetq.c	LSP quantizer
lspgett.c	compute LSP quantizer distortion
lspgetw.c	compute LSP weights
lsplast.c	select LSP MA predictor
lsppre.c	pre-selection first LSP codebook

TABLE 13-continued

Summary of encoder specific routines.

Filename	Description
lspprev.c	LSP predictor routines
lspsel1.c	first stage LSP quantizer
lspsel2.c	second stage LSP quantizer
lspstab.c	stability test for LSP quantizer
pitch_fr.c	closed-loop pitch search
pitch_ol.c	open-loop pitch search
pre_proc.c	pre-processing (HP filtering and scaling)
pwf.c	computation of perceptual weighting coefficients
qua_gain.c	gain quantizer
qua_lsp.c	LSP quantizer
relspwe.c	LSP quantizer

TABLE 14

Summary of decoder specific routines.

Filename	Description
d_lsp.c	decode LP information
de_acelp.c	decode algebraic codebook
dec_gain.c	decode gains
dec_lag3.c	decode adaptive codebook index
dec_ld8k.c	decoder routine
lspdec.c	LSP decoding routine
post_pro.c	post processing (HP filtering and scaling)
pred_lt3.c	generation of adaptive codebook
pst.c	postfilter routines

TABLE 15

Summary of general routines.

Filename	Description
basicop2.c	basic operators
bits.c	bit manipulation routines
gainpred.c	gain predictor
int_lpc.c	interpolation of LSP
inter_3.c	fractional delay interpolation
lsp_az.c	compute LP from LSP coefficients
lsp_lsf.c	conversion between LSP and LSF
lsp_lsf2.c	high precision conversion between LSP and LSF
lspexp.c	expansion of LSP coefficients
lspstab.c	stability test for LSP quantizer
p_parity.c	compute pitch parity
pred_lt3.c	generation of adaptive codebook
random.c	random generator
residu.c	compute residual signal
syn_filt.c	synthesis filter
weight_a.c	bandwidth expansion LP coefficients

The invention claimed is:

1. A method for use in a speech decoder which falls to receive reliably at least a portion of each of first and second consecutive frames of compressed speech information, the speech decoder including a codebook memory for supplying a vector signal in response to a signal representing pitch-period information, the vector signal for use in generating a decoded speech signal, the method comprising:

storing a signal having a value representing pitch-period information corresponding to said first frame; and

incrementing said value of said signal for use in said second frame, such that said codebook memory supplies a vector signal in response to the incremented value of said signal.

2. The method of claim 1 wherein the value of the signal representing pitch-period information is in units of samples of a signal representing speech information.

3. The method of claim 2 wherein the step of incrementing comprises incrementing a number of samples representing a pitch-period.

4. The method of claim 1 wherein the signal value representing pitch-period informtion corresponding to said first frame is equal to a value of pitch-period information received in a frame in which no failure to receive information has occurred.

5. A method for use in a speech decoder which fails to receive reliably at least a portion of a frame of compressed speech information for first and second consecutive frames, the speech decoder including an adaptive codebook memory for supplying codebook vector signals for use in generating a decoded speech signal in response to a signal representing pitch-period information, the method comprising:

storing a signal having a value representing pitch-period information corresponding to said first frame; and

if said stored value does not exceed a threshold, incrementing said value of said signal for use in said second frame.

* * * * *

EXHIBIT C

US006611254B1

(12) **United States Patent**
Griffin et al.

(10) **Patent No.: US 6,611,254 B1**
(45) **Date of Patent: *Aug. 26, 2003**

(54) HAND-HELD ELECTRONIC DEVICE WITH A KEYBOARD OPTIMIZED FOR USE WITH THE THUMBS

(75) Inventors: **Jason T. Griffin**, Waterloo (CA); **John A. Holmes**, Waterloo (CA); **Mihal Lazaridis**, Waterloo (CA); **Herb A. Little**, Waterloo (CA); **Harry R. Major**, Waterloo (CA)

(73) Assignee: **Research In Motion Limited**, Waterloo (CA)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

This patent is subject to a terminal disclaimer.

(21) Appl. No.: **09/634,774**

(22) Filed: **Aug. 9, 2000**

Related U.S. Application Data

(60) Division of application No. 09/106,585, filed on Jun. 29, 1998, which is a continuation-in-part of application No. 29/089,942, filed on Jun. 26, 1998, now Pat. No. Des. 416,256.

(51) **Int. Cl.**^{7} **G09G 5/00**
(52) **U.S. Cl.** **345/169**; 345/168
(58) **Field of Search** 345/156, 157, 345/169, 168, 901, 184; 341/22; 707/4, 526; 709/200, 201, 202, 203, 204, 205, 206, 231, 232; 455/3.01, 3.03, 3.02

(56) **References Cited**

U.S. PATENT DOCUMENTS

D312,628	S		12/1990	Yokoi et al.	
D313,401	S		1/1991	Tanabe	
D313,413	S		1/1991	Langton	
5,410,141	A		4/1995	Koenck et al.	
5,426,449	A	*	6/1995	Danziger	400/472
D359,920	S		7/1995	Sakamoto	
5,457,454	A	*	10/1995	Sugano	341/22
5,500,643	A	*	3/1996	Grant	341/22
5,563,631	A	*	10/1996	Masunaga	345/169
5,606,712	A		2/1997	Hidaka	
5,611,031	A		3/1997	Hertzfeld et al.	
D381,021	S		7/1997	Williams et al.	
D383,756	S	*	9/1997	Henderson et al.	D14/191
D390,509	S		2/1998	Antzinas et al.	
5,818,437	A		10/1998	Grover et al.	
5,825,353	A	*	10/1998	Will	345/184
D402,572	S		12/1998	Han	
5,915,228	A		6/1999	Kunihiro et al.	
5,995,026	A	*	11/1999	Sellers	341/34
D420,351	S		2/2000	Waldner	
6,023,779	A	*	2/2000	Fullam et al.	714/751
6,049,796	A	*	4/2000	Siitonen et al.	707/3
6,052,070	A		4/2000	Kivela et al.	
6,091,956	A	*	7/2000	Hollenberg	455/456
6,094,197	A	*	7/2000	Buxton et al.	345/358
6,102,594	A		8/2000	Strom	
6,103,979	A		8/2000	Motoyama et al.	
6,107,997	A	*	8/2000	Ure	345/173
D432,511	S		10/2000	Eckholm	

(List continued on next page.)

FOREIGN PATENT DOCUMENTS

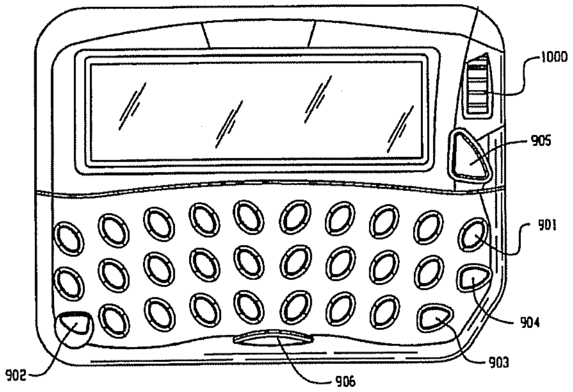
EP	0760291		3/1997	
EP	1143327 A1	*	10/2001	G06F/3/023
WO	WO 99/37025	*	7/1999	H03K/17/94
WO	WO 00/30381	*	5/2000	H04Q/7/22

Primary Examiner—Vijay Shankar
Assistant Examiner—Mansour M. Said
(74) *Attorney, Agent, or Firm*—Jones Day; Krishna K. Pathiyal, Esq.; Charles B. Meyer, Esq.

(57) ABSTRACT

A hand-held electronic device with a keyboard optimized for use with the thumbs is disclosed. In order to operate within the limited space available on a hand-held electronic device, the present invention optimizes the placement and shape of the keys, preferably using keys that are oval or oblong in shape, and that are placed at angles designed to facilitate thumb-typing. The angles at which keys on either side of the keyboard are placed is complimentary.

25 Claims, 4 Drawing Sheets



U.S. PATENT DOCUMENTS

D433,017	S		10/2000	Martinez	
6,148,261	A	*	11/2000	Obradovich et al.	701/208
6,157,323	A	*	12/2000	Tso et al.	341/22
D436,591	S		1/2001	Abston et al.	
6,212,412	B1	*	4/2001	Rogers et al.	455/575
D441,733	S		5/2001	Do et al.	
6,243,789	B1	*	6/2001	Hasbun et al.	711/103
D451,079	S		11/2001	Ali	
D454,349	S		3/2002	Makidera et al.	
D454,849	S		3/2002	Eckholm	
6,374,277	B2	*	4/2002	Vong et al.	708/112
D456,794	S		5/2002	Laverick et al.	
D458,239	S		6/2002	Shim et al.	
D459,327	S		6/2002	Ali	
D460,068	S		7/2002	Lanzaro et al.	
6,459,968	B1	*	10/2002	Kochie	701/29

* cited by examiner

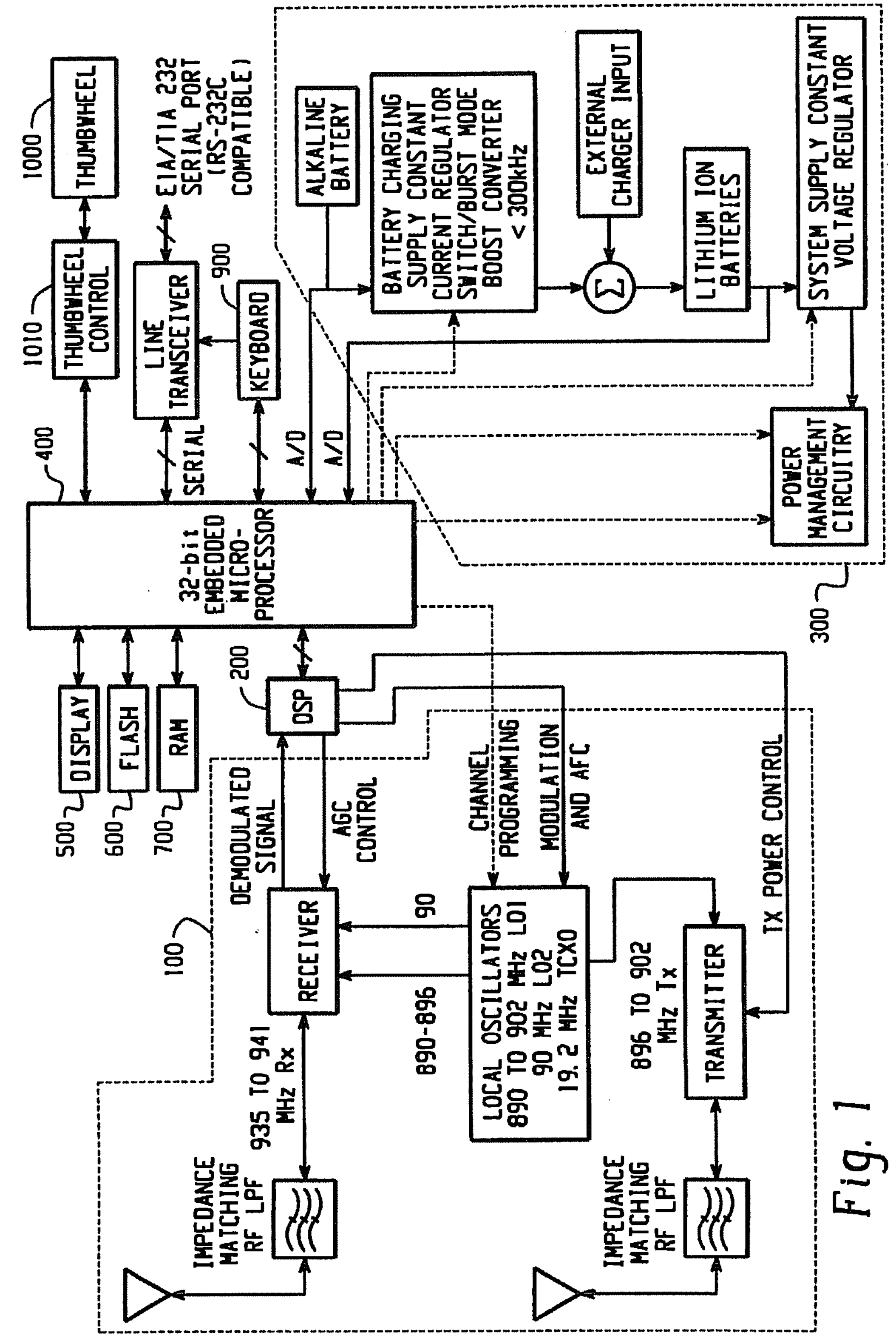

1000
THUMBWHEEL
1010
THUMBWHEEL CONTROL
EIA/TIA 232 SERIAL PORT (RS-232C COMPATIBLE)
LINE TRANSCEIVER
900
KEYBOARD
ALKALINE BATTERY
BATTERY CHARGING SUPPLY CONSTANT CURRENT REGULATOR SWITCH/BURST MODE BOOST CONVERTER < 300kHz
EXTERNAL CHARGER INPUT
Σ
LITHIUM ION BATTERIES
SYSTEM SUPPLY CONSTANT VOLTAGE REGULATOR
POWER MANAGEMENT CIRCUITRY
300
400
SERIAL
A/D
A/D
32-bit EMBEDDED MICRO-PROCESSOR
DISPLAY
FLASH
RAM
500
600
700
200
DSP
DEMODULATED SIGNAL
AGC CONTROL
CHANNEL PROGRAMMING
MODULATION AND AFC
TX POWER CONTROL
100
RECEIVER
90
890-896
LOCAL OSCILLATORS 890 TO 902 MHz LO1 90 MHz LO2 19.2 MHz TCXO
896 TO 902 MHz Tx
TRANSMITTER
935 TO 941 MHz Rx
IMPEDANCE MATCHING RF LPF
IMPEDANCE MATCHING RF LPF

Fig. 1

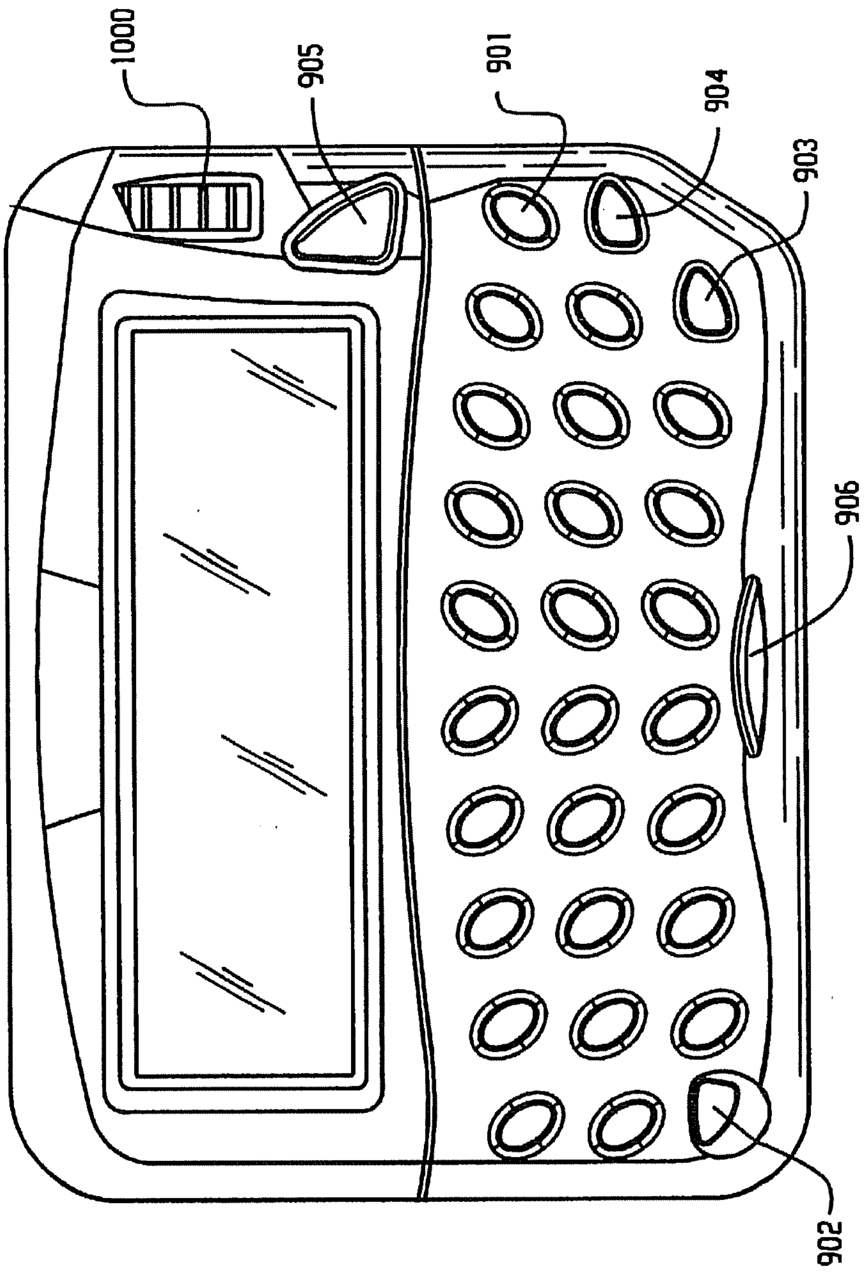


Fig. 2

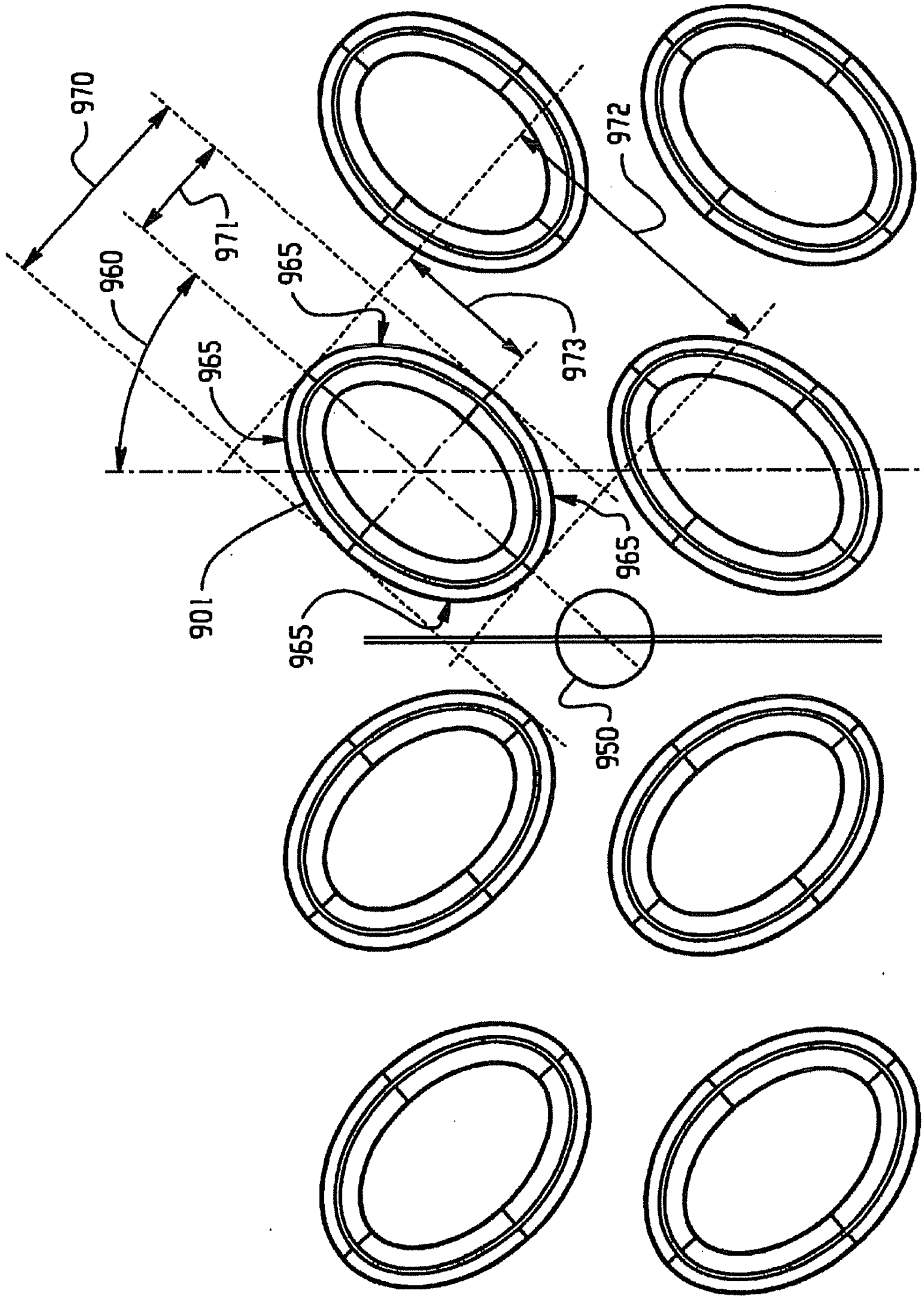


Fig. 3

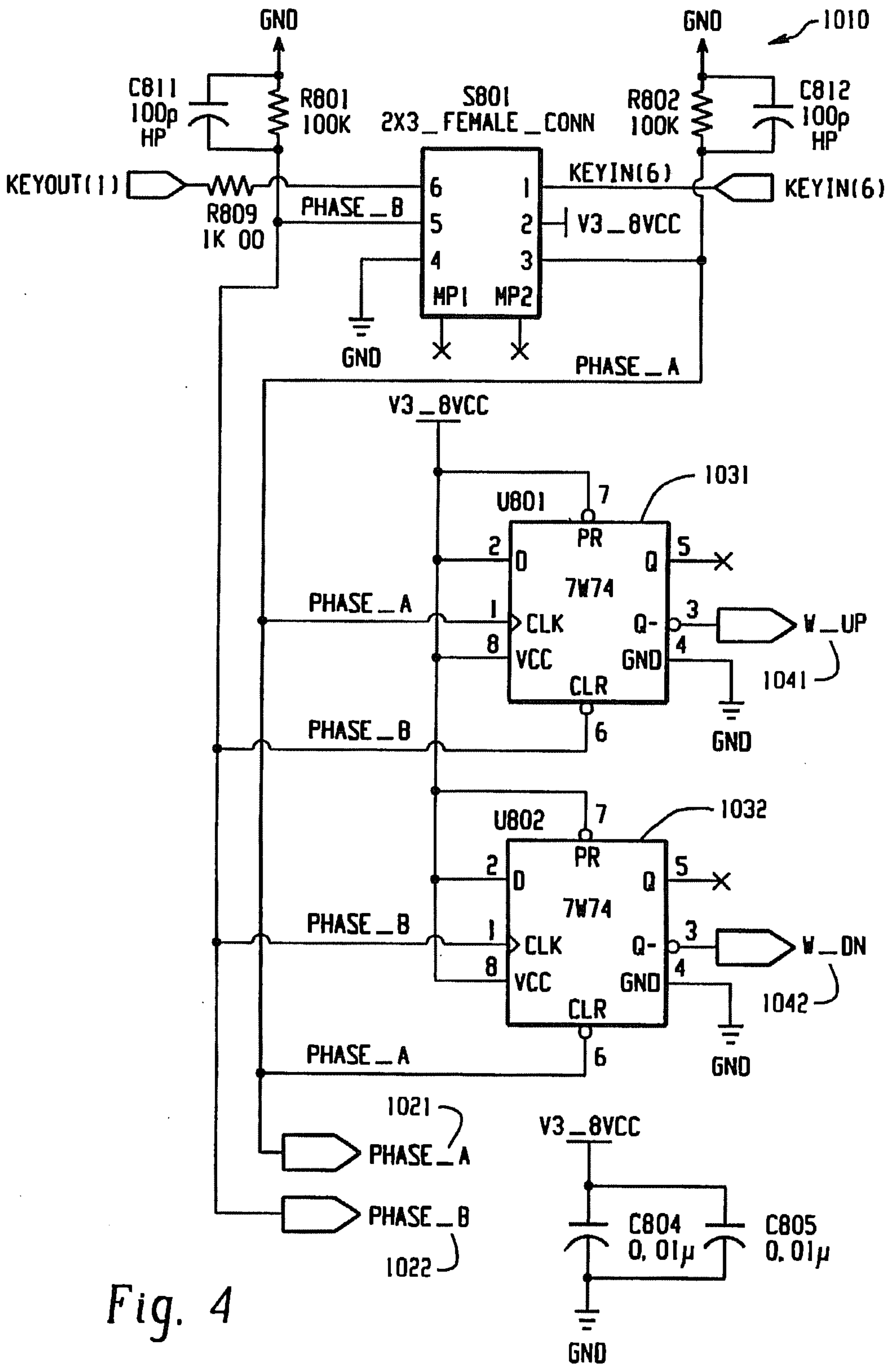

GND
GND
1010
C811
100p
HP
R801
100K
S801
2X3_FEMALE_CONN
R802
100K
C812
100p
HP
KEYOUT(1)
R809
1K 00
PHASE_B
6 1
5 2
4 3
MP1 MP2
KEYIN(6)
V3_8VCC
KEYIN(6)
GND
PHASE_A
V3_8VCC
U801
7
1031
PR
2
D
Q
5
7W74
PHASE_A
1
CLK
Q-
3
W_UP
8
VCC
GND
4
1041
CLR
PHASE_B
6
GND
U802
7
1032
PR
2
D
Q
5
7W74
PHASE_B
1
CLK
Q-
3
W_DN
8
VCC
GND
4
1042
CLR
PHASE_A
6
GND
1021
PHASE_A
V3_8VCC
PHASE_B
C804
0.01μ
C805
0.01μ
1022
GND

Fig. 4

HAND-HELD ELECTRONIC DEVICE WITH A KEYBOARD OPTIMIZED FOR USE WITH THE THUMBS

This application is a Division of U.S. Ser. No. 09/106, 585, entitled "Hand-Held Electronic Device With A Keyboard Optimized For Use With The Thumbs", filed Jun. 29, 1998, which is a continuation-In-Part of U.S. Design Application Ser. No. 29/089,942 entitled "Hand-Held Messaging Device With Keyboard", filed Jun. 26, 1998, U.S. Pat. Des. 416,256 and assigned to the assignee of the present invention.

BACKGROUND OF THE INVENTION

The present invention is directed toward the field of small, hand-held electronic devices such as personal data assistants (PDA's), personal information managers (PIM's), two-way pagers and the like. In particular, the system and method of the present invention provide the user of the hand-held device with the ability to input data with a minimal amount of key strokes and optimized for use substantially with the thumbs.

In a two-way paging system that provides two-way, full text messaging, there is a need to permit the user to initiate messages and to respond to messages in a timely fashion and with text entirely created by the user. In order to keep the form factor of the two-way pager small enough to be worn on the body of the user, such as with a belt clip, the input device needs to be small, have a minimal number of keys and optimized for use with a minimal number of key strokes. Prior art systems have attempted to address these needs by incorporating virtual keyboards or pen-based input systems for user inputs to the device, but such systems require the user to input data in an unfamiliar manner. Additionally, in a small hand-held messaging device, such as a two-way pager, these systems prove awkward to use.

In order to provide a hand-held electronic device that permits a user the opportunity to enter data into an address book, a calendar, a task list, an email message or a similar text file that requires user-generated data, the instant invention is directed to an input device that is oriented to be used substantially through use of the thumbs. This is accomplished first by providing a keyboard with a minimal number of keys, but with the keys representing the alphabet generally placed in the same order as they would appear on a standard keyboard, such as in a standard QWERTY or a DVORAK keyboard layout. The use of a keyboard layout that is familiar to the user enables the user to immediately use the device without having to hunt for the keys he or she wishes to use.

Although the layout is similar to a standard keyboard, the keys are placed at an orientation and in a particular shape that attempts to maximize the surface area of the thumb hitting the key and to provide the user with a comfortable position of the hands for data input. Also, the orientation encourages input by the thumbs, which the inventors of the instant invention have discovered to be faster and more accurate in small hand-held electronic devices than touch-typing or "hunting and pecking" typing.

An additional feature of the invention is the use of an additional input means for control of functions that might otherwise be controlled by a keyboard that included function keys. To encourage data entry using thumbs and again to minimize the number of keys on the keyboard, the instant invention also includes a thumb-wheel for control of menus for selection of forms and functions relevant to data input.

The thumb-wheel is positioned in close proximity to the keyboard to enable the easily transition from thumb-based typing to thumb control of forms and functions.

In addition to hardware features that encourage optimal data entry through the use of thumbs, there are several software features that are designed to minimize keystrokes and aid in entry of data.

The features of this invention, both individually and collectively, have not, to the knowledge of the inventors, been applied to a small hand-held electronic device that requires user-generated data entry. To permit efficient operation of such devices while keeping the form factor of the device small enough to be worn on the body, there is a general need for a hand-held electronic device that can fit in the palm of the hand and that can be operated substantially with the thumbs.

There is a further need for a keyboard for a palm-size data entry device with keys placed at an angle to optimize operation of the keyboard by the use of the thumbs.

There remains another need for a keyboard with keys that are shaped and sized to maximize contact with the thumbs while minimizing the keyboard area required for such keys.

There also remains a need for an auxiliary input device that is to be operated by the thumb for data inputs forms and function control and that, in conjunction with the keyboard, encourages and permits data entry and management through input performed substantially by the thumbs.

There remains still another need for a software-implemented user interface system that is designed, at least in part, to support and encourage data entry through use of the thumbs.

SUMMARY OF THE INVENTION

The present invention overcomes the problems noted above and satisfies the needs in this field for a hand-held electronic device with a keyboard optimized for use with the thumbs. In the preferred embodiment of the present invention, the hand-held electronic device is a two-way paging device that permits full-text, two-way messaging such as email messaging and that includes standard PDA or PIM features such as an address book, an electronic calendar, a task list and other text-based features. These features require user input of text strings that can be lengthy and that cannot be reduced to pre-determined or "canned" strings. Thus, for such a device, the efficient entry of data in a device meant to fit into the palm of one's hand requires that two goals are achieved. First, the data entry must be relatively easy from a user perspective. This means that the user must be somewhat familiar with analogous forms of data entry and not have to be trained to use the data entry for the hand-held device. Second, the form factor does not permit a large number of keys or keys that are very large. Thus efficient use of the keyboard space is required and functions that might be able to be performed by a standard keyboard are off-loaded to an auxiliary input device or are performed, through a minimal number of keystrokes that encourage the use of thumb-based data entry.

To accomplish these goals, the invention first optimizes the placement of the keys on the device keyboard. In order to work within the limited space available for the keyboard, it was determined that it was preferable to use keys that were oval or oblong and that were placed at angles designed to facilitate use by thumb typing. An angle for the keys on the right side of the keyboard and a complementary angle for the keys on the left side of the keyboard are chosen based upon observation of the angle at which a user will orient his or her thumbs while thumb-typing.

The invention also minimizes the number of keys available for data input. In the preferred embodiment, only keys for the 26 letters of the English alphabet are available as well as a backspace key, a line feed key, an "alt" key, a "cap" key and a space bar. The alt key enables the user in conjunction the other keys to input numbers and symbols to perform certain functions. The placement of the keys is designed to enhance the user experience while typing with the thumbs by meeting two seemingly opposite goals —minimizing the keyboard footprint while maximizing the likelihood that proper keys will be struck by the thumb-typing user.

The invention also provides additional incentive for the user to use thumb input by providing an input device adjacent to the keyboard, but integral to the overall handheld device. Although other devices can be used in an auxiliary fashion, the preferred device is a thumbwheel that registers movement of the wheel by measuring the number of indents traversed while rolling the wheel and that also registers as an input the depression or "clicking" of the wheel, which is performed by pressing the wheel toward the back of the pager. This clicking of the wheel is similar to the clicking of a mouse associated with a PC or any other input device that registers the depression of a button. The thumbwheel in the preferred embodiment is placed vertically on the two-way paging device so that the user can easily move his or her thumb from the thumbwheel to the keyboard and back for performing functions and retrieving data forms, such as an e-mail template or address book entry template, for data entry.

Additionally, various software techniques can be implemented to enhance the thumb-typing user's experience in using the device of the instant invention. In the preferred embodiment, for example, the user can change the capitalization of a particular letter simply by keeping a key depressed for a particular length of time without an intermittent release being detected by the keyboard controller.

The primary advantage of the present invention is that it enables efficient and user-friendly data entry into a palm-sized electronic device by maximizing the potential for user data entry through thumb typing.

These are just a few of the many advantages of the present invention, as described in more detail below. As will be appreciated, the invention is capable of other and different embodiments and its several details are capable of modifications in various respects, all without departing from the spirit of the invention. Accordingly, the drawings and description of the preferred embodiment set forth below are to be regarded as illustrative in nature and not restrictive.

BRIEF DESCRIPTION OF THE DRAWINGS

The present invention satisfies the needs noted above as will become apparent from the following description when read in conjunction with the accompanying drawings wherein:

FIG. 1 is a block diagram of a two-way, full-text, messaging device incorporating a keyboard and an auxiliary data entry device.

FIG. 2 is a frontal view of the hand-held device showing the shape and placement of the keys on the keyboard and the auxiliary input device.

FIG. 3 is a diagram of showing the shape, size and placement of the keys on the keyboard.

FIG. 4 is a diagram of the control circuitry for the thumbwheel.

DETAILED DESCRIPTION OF THE INVENTION

Referring now to the drawings, FIG. 1 is a block diagram of the major subsystems and elements comprising a palm-

sized, mobile, two-way messaging device that preferably incorporates the invention. In its broadest terms, the messaging device includes a transmitter/receiver subsystem **100** connected to a DSP **200** for digital signal processing of the incoming and outgoing data transmissions, power supply and management subsystem **300**, which supplies and manages power to the overall messaging device components, microprocessor **400**, which is preferably an X86 architecture processor, that controls the operation of the messaging device, display **500**, which is preferably a full graphic LCD, FLASH memory **600**, RAM **700**, serial output and port **800**, keyboard **900** , thumbwheel **1000** and thumbwheel control logic **1010**. In its intended use, a message comes via a wireless data network, such as the Mobitex network, into subsystem **100**, where it is demodulated via DSP **200** and decoded and presented to microprocessor **300** for display on display **500**. To access the display of the message, the user may choose from functions listed under a menu presented as a result of user interaction with thumbwheel **1000**. If the message is an email message, the user may chose to respond to the email by selecting "Reply" from a menu presented on the display through interaction via thumbwheel **1000** or via menu selection from keyboard **900**. In typing the reply, the user can use keyboard **900** to type full text message replies, or insert pre-determined or "canned" response by using either a particular keystroke pattern or through pulling down pre-determined text strings from a menu of items presented on display **500** through the use of thumbwheel **1000**. When the reply to the message is composed, the user can initiate the sending of the message preferably by interaction through thumbwheel **1000**, or alternatively, with less efficiency. through a combination of key board **900** keystrokes. When the microprocessor **300** receives an indication that the message is to be sent, it processes the message for transport and, by directing and communicating with transmitter/receiver subsystem **100**, enables the reply message to be sent via the wireless communications data network to the intended recipient. Similar interaction through I/O devices keyboard **900** and thumbwheel **1000** can be used to initiate full-text messages or to forward messages to another party. Also, the keyboard **900** and thumbwheel **1000** can be used to permit data entry to an address book resident on the messaging device, or an electronic calendar or log book, or any other function on the messaging device requiring data entry. Preferably, the thumbwheel is a thumbwheel with a push button SPST with quadrature signal outputs, such as that manufactured by Matsushita Electronic Components Co. Ltd. as part number EVQWK2001.

FIG. **2** is a front view of messaging device **10** that incorporates the invention. Shown in FIG. **2** are a plurality of letter keys **901**, and specialized keys **902**, **903**, **904** and **905** and space bar **906**. Also shown is thumbwheel **1000** in its vertical orientation and in association with display **500** and keyboard **900**. In the preferred embodiment, **902** is the alt key, **903** is the cap key, **904** is the line feed key and **905** is the backspace key.

FIG. **3** is a view of a subset of the letter keys **901**, showing the dimensions and relative position of the keys. Shown also is the point **950** that marks the center of keyboard **900**, key dimensions **970**, **971**, **972** and **973**, as well as angle **960** and the rho value **965**, representing curvature of a letter key **901**. In investigating optimal key placement on the keyboard, it was determined that the keys should be placed at an angle **960** relative to vertical that facilitated easy typing using thumbs. That angle is preferably positive 40 degrees relative to vertical for keys on the right side of the keyboard (where **950** is the center of the keyboard) and negative 40 degrees for the keys on the left side of the keyboard, although complementary angles ranging from 20 degrees to 70 degrees could also be used to accomplish the goal, albeit less optimally, of facilitating thumb typing. Also as shown on FIGS. **2** and **3**, the keys are dispersed across keyboard **900** evenly so that there is sufficient space between the keys to decrease the opportunity for multiple keys being depressed while thumb typing. Additionally, the keys are sized appropriately given the footprint of the messaging device and the keyboard **900**. In its preferred embodiment, the messaging device **10** measures across its face 64 mm by 89mm, which does not leave much room for keyboard **900** and display **500**. In the preferred embodiment, keyboard **900** occupies over half of the face of the messaging device **10**.

The key shape and dimensions are also key components of the invention. In order to maximize the surface area of the key that a thumb would hit, the keys are preferably oval, and have a rho **965** defining the curvature of the key of **414**, although values may range higher or lower. Other rho values will lead to an acceptable, but not as optimal or aesthetically pleasing shape of keys **901**. As to the key dimensions, the width **970** of the key **901** is 4.8 millimeters (**971** representing the radius of half that value, 2.4 mm) and the length (or height) **972** of the key **901** is 7 millimeters millimeters (**973** representing the radius of half that value, 3.5 mm).

Turning to one of the software features that aids in the device **10** being optimally used for thumb typing is a capitalization feature implemented via software. If a user depresses a key **901**, the operating system detects a key down event. If the key is released after a period of time, the operating system detects a key up event. If upon a key down event, a period of time elapses before a key up event is detected, the operating system determines that a key repeat event has occurred representing a situation where a user has continued to depress a key without releasing it. A key repeat event is then treated by application software residing in either FLASH **600** or RAM **700** as an event that requires the capitalization of the key previously depressed. This feature disables a key repeat feature and substitutes instead a capitalization feature based upon a key repeat. The timing of the key scanning to determine whether a key has been released can be set to permit a slower keyboard response or a faster keyboard response, depending upon user experience or preferences. Although the capitalization function preferably works only to change the state of a letter to a capital, it alternatively could operate to change a capital letter to a lower case letter. The actual display is changed by the application program substituting the value of the capital letter in the register that holds the value of the letter to be displayed. As alternatively implemented, the continued depressing without release of a letter key could result in a key oscillating between upper case and lower case, depending on the length of time the key is depressed.

FIG. **4** is the logic circuitry **1010** associated with thumbwheel **1000**. Thumbwheel **1000** outputs quadrature signals phase A **1021** and phase B **1022**, which are processed by D flip-flops **1031** and **1032** to present signals **1041** W_{13}UP and **1042** W_{13}DN to microprocessor **300**. Signals **1041** and **1042** represent, respectively, a user rolling the thumbwheel up and rolling the thumbwheel down.

Having described in detail the preferred embodiments of the present invention, including the preferred methods of operation, it is to be understood that this operation could be carried out with different elements and steps. This preferred embodiment is presented only by way of example and is not meant to limit the scope of the present invention which is defined by the following claims.

What is claimed:

1. A method in a mobile communication device having a wireless transceiver, a display, a QWERTY-style keyboard, and an auxiliary input device, the method comprising the steps of:

receiving a plurality of e-mail messages with the wireless transceiver, the plurality of e-mail messages being transmitted from a wireless network;

displaying a list of the plurality of received e-mail messages on the display;

selecting and accessing one of the plurality of received e-mail messages by moving the auxiliary input device in a direction that corresponds to a desired direction on the displayed list and by depressing the auxiliary input device to select and access a particular received e-mail message;

displaying the selected message on the display;

activating a menu of e-mail commands and displaying the menu of commands on the display;

selecting and executing one of the e-mail commands on the menu by moving the auxiliary input device in a direction that corresponds to a desired direction on the displayed menu and by depressing the auxiliary input device to select and execute a particular e-mail command, wherein one of the e-mail commands is a command to generate a reply message to the displayed message;

if the command to generate a reply message is selected and executed, then (a) displaying a reply e-mail message form on the display; (b) inputting reply message text using the QWERTY-style keyboard; and (c) selecting and executing a send e-mail command from a menu of commands displayed on the display using the auxiliary input device to transmit the reply message from the mobile communications device to the wireless network using the wireless transceiver.

2. The method of claim **1**, wherein the mobile communication device further comprises a microprocessor coupled to a memory for storing an operating system that is executed by the microprocessor, the operating system enabling the steps of:

displaying a first character on the display when a key is selected using the QWERTY-style keyboard; and

displaying a second character on the display in place of the first character when the key is continually depressed for a predetermined period of time.

3. The method of claim **2**, wherein the first character is a lower case character and the second character is an upper case character.

4. The method of claim **2**, wherein the predetermined period of time can be modified by a user of the electronic device.

5. The method of claim **1**, wherein the QWERTY-style keyboard includes a plurality of oblong shaped letter keys.

6. The method of claim **5**, wherein the plurality of oblong shaped letter keys are oval shaped keys.

7. The method of claim **1**, wherein the mobile communication device is a two-way pager, a personal digital assistant, or a cellular telephone.

8. The method of claim **1**, wherein the auxiliary input device is a thumbwheel.

9. The method of claim **1**, further comprising the steps of:

storing a plurality of application programs in a memory of the mobile communication device;

displaying a list of the plurality of application programs on the display;

selecting and executing one of the plurality of application programs by moving the auxiliary input device in a direction that corresponds to a desired direction on the displayed list and by depressing the auxiliary input device to select and execute a particular one of the plurality of application programs.

10. The method of claim **9**, wherein the plurality of application programs include an e-mail application and a calendar application.

11. The method of claim **10**, wherein the plurality of application programs further include an address book application.

12. An electronic wireless messaging device, comprising;

a microprocessor;

a transceiver coupled to the microprocessor for sending and receiving electronic messages via a wireless network;

a display coupled to the microprocessor for displaying a list of received electronic messages from the wireless network and for displaying a menu of electronic message processing commands;

a QWERTY-style keyboard coupled to the microprocessor for inputting electronic messages; and

an auxiliary input device for selecting and accessing one of the received electronic messages displayed on the list of received electronic messages by moving the auxiliary input device in a direction that corresponds to a desired direction on the displayed list and by depressing the auxiliary input device to select and access a particular received electronic message;

wherein the auxiliary input device is also for selecting and executing one of the electronic message processing commands on the menu by moving the auxiliary input device in a direction that corresponds to a desired direction on the displayed menu and by depressing the auxiliary input device to select and execute a particular electronic message processing command.

13. The electronic wireless messaging device of claim **12**, wherein one of the electronic message processing commands is a generate reply message command, which when accessed and executed by the auxiliary input device causes the electronic messaging device to display a reply electronic message form, wherein a user of the electronic messaging device inputs reply message text using the keyboard into the reply electronic message form, and then selects and executes a send electronic message command from the menu of electronic message processing commands to transmit the reply message from the electronic wireless messaging device to the wireless network using the transceiver.

14. The electronic wireless messaging device of claim **12**, wherein the QWERTY-style keyboard comprises a plurality of oblong shaped keys.

15. The electronic wireless messaging device of claim **14**, wherein the plurality of oblong shaped keys are oval shaped.

16. The electronic wireless messaging device of claim **14**, wherein the plurality of oblong shaped keys are tilted at an angle with respect to a vertical reference line through the electronic wireless messaging device.

17. The electronic wireless messaging device of claim **16**, wherein the plurality of oblong shaped keys comprise a first set of oblong shaped keys and a second set of oblong shaped keys, wherein the first set of oblong shaped keys are tilted at a positive angle with respect to the vertical reference line and the second set of oblong shaped keys are tilted at a negative angle with respect to the vertical reference line.

18. The electronic wireless messaging device of claim **12**, further comprising a memory coupled to the microprocessor for storing an operating system that is executed by the microprocessor, the operating system enabling the steps of:

displaying a first character on the display when a key is selected using the QWERTY-style keyboard; and

displaying a second character on the display in place of the first character when the key is continually depressed for a predetermined period of time.

19. The electronic wireless messaging device of claim **18**, wherein the first character is a lower case character and the second character is an upper case character.

20. The electronic wireless messaging device of claim **18**, wherein the predetermined period of time can be modified by a user of the electronic device.

21. The electronic wireless messaging device of claim **12**, wherein the electronic wireless messaging device is a two-way pager, a personal digital assistant, or a cellular telephone.

22. The electronic wireless messaging device of claim **12**, wherein the auxiliary input device is a thumbwheel.

23. The electronic wireless messaging device of claim **12**, further comprising:

a memory for storing a plurality of application programs;

wherein the display displays a list of the plurality of application programs, and wherein the auxiliary input device is used to select and execute one of the plurality of application programs by moving the auxiliary input device in a direction that corresponds to a desired direction on the displayed list and by depressing the auxiliary input device to select and execute a particular one of the plurality of application programs.

24. The electronic wireless messaging device of claim **23**, wherein the plurality of application programs include an e-mail application and a calendar application.

25. The electronic wireless messaging device of claim **24**, wherein the plurality of application programs further include an address book application.

* * * * *

EXHIBIT D

US006611255B2

(12) **United States Patent**
Griffin et al.

(10) **Patent No.: US 6,611,255 B2**
(45) **Date of Patent: *Aug. 26, 2003**

(54) **HAND-HELD ELECTRONIC DEVICE WITH A KEYBOARD OPTIMIZED FOR USE WITH THE THUMBS**

(75) Inventors: **Jason T. Griffin**, Waterloo (CA); **John A. Holmes**, Waterloo (CA); **Mihal Lazaridis**, Waterloo (CA); **Herb A. Little**, Waterloo (CA); **Harry R. Major**, Waterloo (CA)

(73) Assignee: **Research In Motion Limited**, Waterloo (CA)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

This patent is subject to a terminal disclaimer.

(21) Appl. No.: **10/167,202**

(22) Filed: **Jun. 11, 2002**

(65) **Prior Publication Data**

US 2002/0158847 A1 Oct. 31, 2002

Related U.S. Application Data

(63) Continuation of application No. 09/900,585, filed on Jul. 6, 2001, now Pat. No. 6,452,588, which is a continuation of application No. 09/106,585, filed on Jun. 29, 1998, now Pat. No. 6,278,442, which is a continuation-in-part of application No. 29/089,942, filed on Jun. 26, 1998, now Pat. No. Des. 416,256.

(51) **Int. Cl.**^{7} **G09G 5/00**
(52) **U.S. Cl.** **345/169**; 345/168; 400/489
(58) **Field of Search** 345/156, 157, 345/168, 169, 140, 160, 170, 171; 400/489, 472, 486, 479; 341/20, 22, 21, 23

(56) **References Cited**

U.S. PATENT DOCUMENTS

D278,341	S	4/1985	Scheid
D293,241	S	12/1987	Wan et al.
D312,628	S	12/1990	Yokoi et al.
D313,401	S	1/1991	Tanabe
D313,413	S	1/1991	Langton
5,184,830	A	2/1993	Okada et al.

(List continued on next page.)

FOREIGN PATENT DOCUMENTS

EP	0760291	3/1997
WO	9937025	7/1999
WO	0030381	5/2000

OTHER PUBLICATIONS

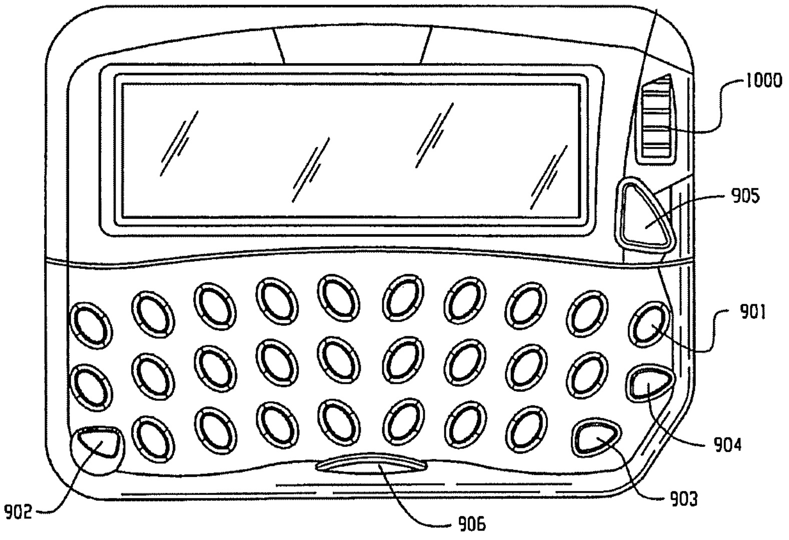
"Programmable Calculators: Hewlett–Packard HP–200LX," Viktor T. Toth, copyr. 2001, 2002, from web page at www.rskey.org/hp2001x.htm.

Primary Examiner—Vijay Shankar
Assistant Examiner—Mansour M. Said
(74) *Attorney, Agent, or Firm*—Jones Day; Krishna K. Pathiyal, Esq.

(57) **ABSTRACT**

A hand-held electronic device with a keyboard optimized for use with the thumbs is disclosed. In order to operate within the limited space available on a hand-held electronic device, the present invention optimizes the placement and shape of the keys, preferably using keys that are oval or oblong in shape, and that are placed at angles designed to facilitate thumb-typing.

68 Claims, 4 Drawing Sheets



U.S. PATENT DOCUMENTS

5,217,295	A		6/1993	Tortola et al.	
5,367,298	A	*	11/1994	Axthelm	341/22
D357,253	S		4/1995	Wong	
5,410,141	A		4/1995	Koenck et al.	
5,426,449	A		6/1995	Danziger	
D359,920	S		7/1995	Sakamoto	
5,436,954	A		7/1995	Nishiyama et al.	
5,457,454	A		10/1995	Sugano	
5,500,643	A	*	3/1996	Grant	341/22
5,606,712	A		2/1997	Hidaka	
5,611,031	A		3/1997	Hertzfeld et al.	
D381,021	S		7/1997	Williams et al.	
D383,756	S	*	9/1997	Henderson et al.	D14/191
5,672,108	A		9/1997	Lam et al.	
D386,497	S		11/1997	Huslig et al.	
D390,509	S		2/1998	Antzinas et al.	
5,737,394	A		4/1998	Anderson et al.	
D397,369	S		8/1998	Rissman	
D397,728	S		9/1998	Yuen et al.	
D399,537	S		10/1998	Chi et al.	
5,818,437	A		10/1998	Grover et al.	
5,825,353	A	*	10/1998	Will	345/184
D402,572	S		12/1998	Han	
D403,362	S		12/1998	Fai	
5,893,798	A		4/1999	Stambolic et al.	
5,915,228	A		6/1999	Kunihiro et al.	
5,920,308	A		7/1999	Kim	
5,931,873	A		8/1999	Cisar	
5,963,197	A		10/1999	Bacon et al.	
5,974,238	A		10/1999	Chase, Jr.	
5,982,520	A		11/1999	Weiser et al.	
5,995,026	A		11/1999	Sellers	
6,006,351	A		12/1999	Peretz et al.	
6,014,429	A		1/2000	LaPorta et al.	
D420,351	S		2/2000	Waldner	
6,023,779	A		2/2000	Fullam et al.	
6,047,047	A		4/2000	Aldridge et al.	
6,049,796	A	*	4/2000	Siitonen et al.	707/3
6,052,070	A		4/2000	Kivelä et al.	
6,094,197	A		7/2000	Buxton et al.	
6,102,594	A		8/2000	Strom	
6,103,979	A		8/2000	Motoyama et al.	
6,107,997	A		8/2000	Ure	
D432,511	S		10/2000	Eckholm	
D433,017	S		10/2000	Martinez	
6,148,261	A		11/2000	Obradovich et al.	
6,157,323	A		12/2000	Tso et al.	
D436,591	S		1/2001	Abston et al.	
6,212,412	B1		4/2001	Rogers et al.	
D441,733	S		5/2001	Do et al.	
6,243,789	B1		6/2001	Hasbun et al.	
6,297,795	B1		10/2001	Kato et al.	
D451,079	S		11/2001	Ali	
D454,349	S		3/2002	Makidera et al.	
D454,849	S		3/2002	Eckholm	
D456,794	S		5/2002	Laverick et al.	
D458,239	S		6/2002	Shim et al.	
D459,327	S		6/2002	Ali	
D460,068	S		7/2002	Lanzaro et al.	

* cited by examiner

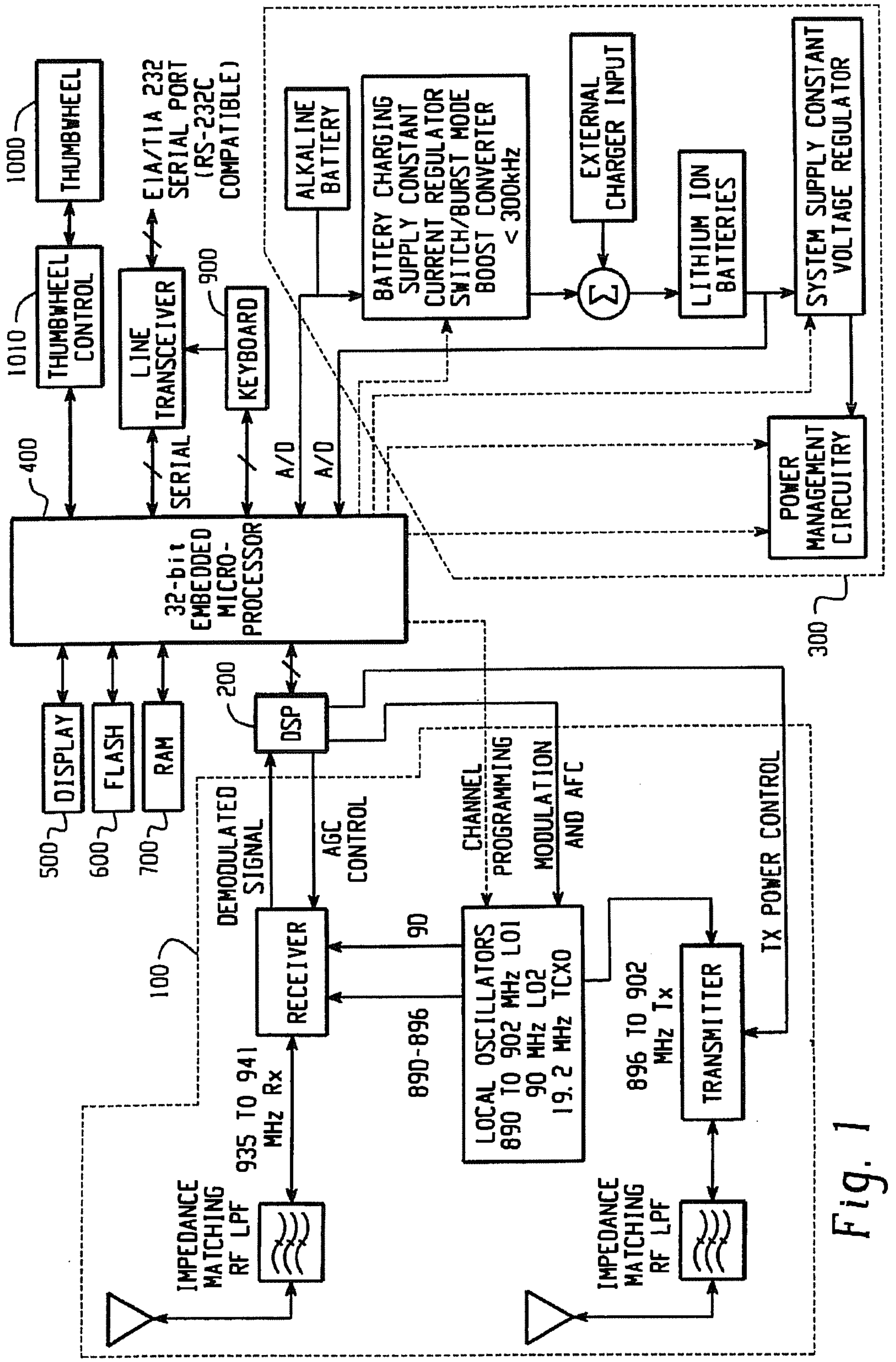

THUMBWHEEL
1000
THUMBWHEEL CONTROL
1010
EIA/TIA 232 SERIAL PORT (RS-232C COMPATIBLE)
LINE TRANSCEIVER
900
KEYBOARD
SERIAL
A/D
A/D
400
32-bit EMBEDDED MICRO-PROCESSOR
ALKALINE BATTERY
BATTERY CHARGING SUPPLY CONSTANT CURRENT REGULATOR SWITCH/BURST MODE BOOST CONVERTER < 300kHz
EXTERNAL CHARGER INPUT
Σ
LITHIUM ION BATTERIES
SYSTEM SUPPLY CONSTANT VOLTAGE REGULATOR
POWER MANAGEMENT CIRCUITRY
300
DISPLAY
500
FLASH
600
RAM
700
DSP
200
DEMODULATED SIGNAL
AGC CONTROL
CHANNEL PROGRAMMING
MODULATION AND AFC
TX POWER CONTROL
100
RECEIVER
90
890-896
LOCAL OSCILLATORS 890 TO 902 MHz LO1 90 MHz LO2 19.2 MHz TCXO
896 TO 902 MHz Tx
TRANSMITTER
935 TO 941 MHz Rx
IMPEDANCE MATCHING RF LPF
IMPEDANCE MATCHING RF LPF

Fig. 1

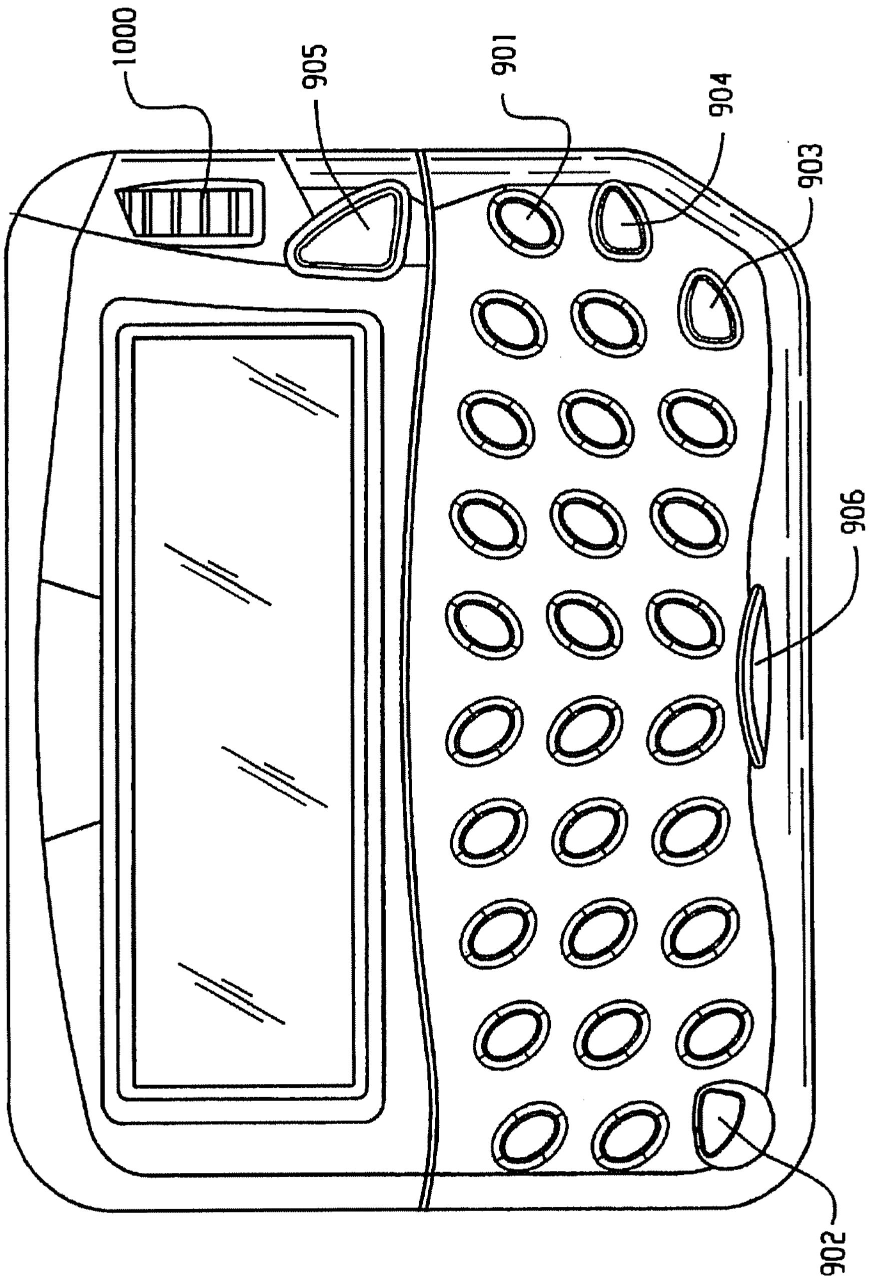


Fig. 2

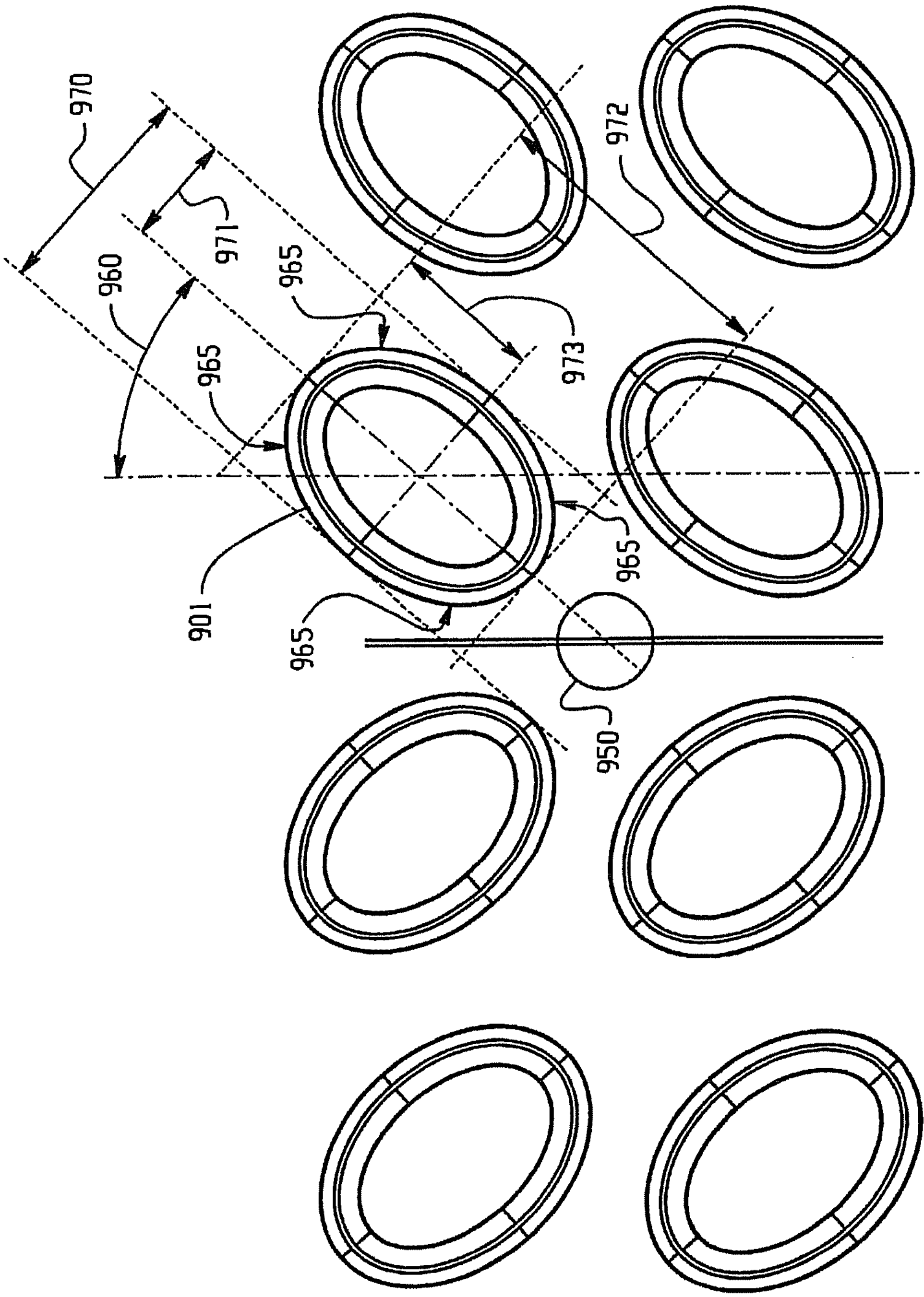


Fig. 3

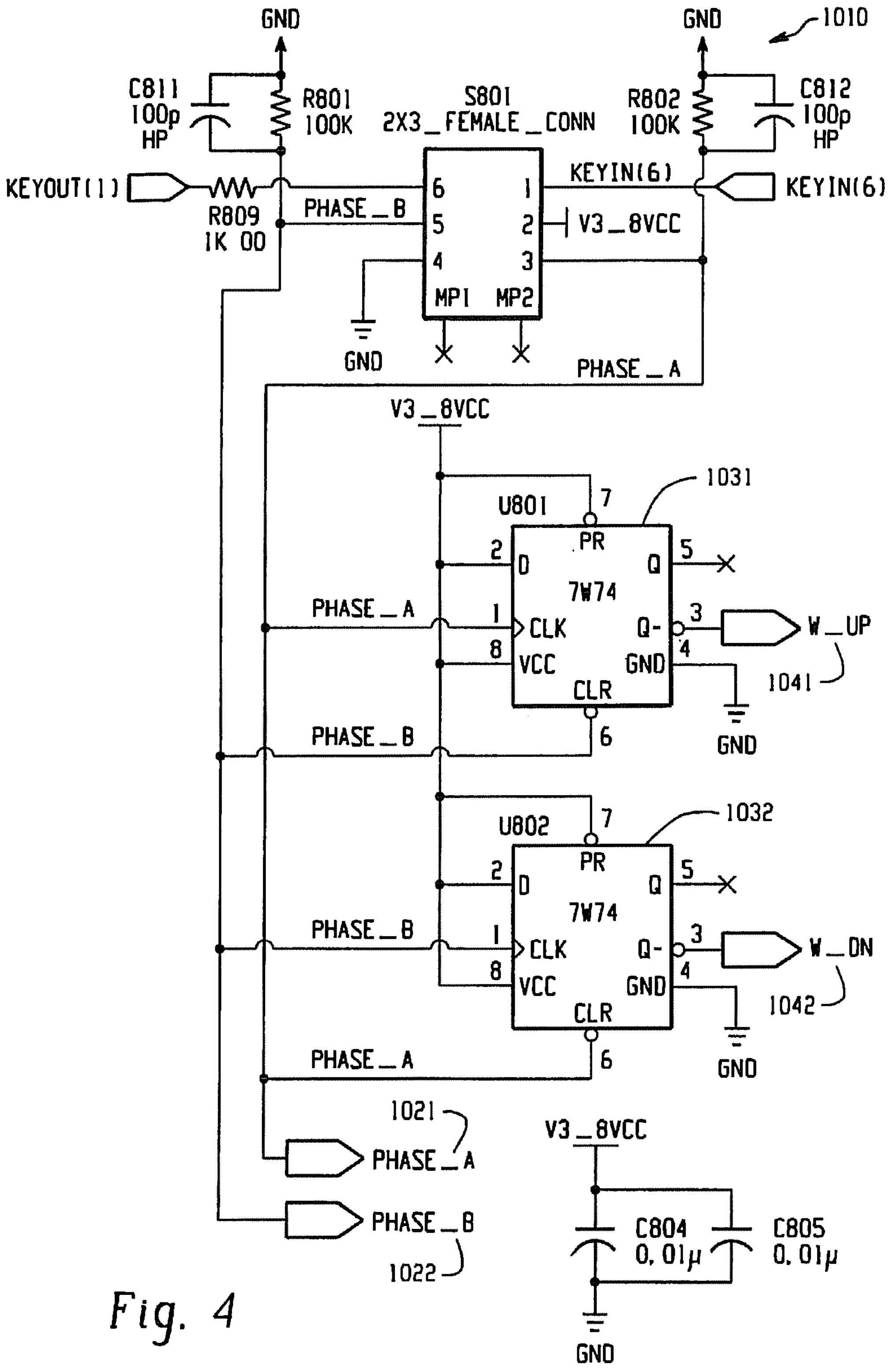

GND
GND
1010
C811
100p
HP
R801
100K
S801
2X3_FEMALE_CONN
R802
100K
C812
100p
HP
KEYOUT(1)
R809
1K 00
PHASE_B
6
5
4
1
2
3
MP1
MP2
KEYIN(6)
KEYIN(6)
V3_8VCC
GND
PHASE_A
V3_8VCC
1031
U801
7
PR
2
D
Q
5
7W74
PHASE_A
1
CLK
Q-
3
W_UP
8
VCC
GND
4
1041
CLR
PHASE_B
6
GND
U802
7
1032
PR
2
D
Q
5
7W74
PHASE_B
1
CLK
Q-
3
W_DN
8
VCC
GND
4
1042
CLR
PHASE_A
6
GND
1021
PHASE_A
V3_8VCC
PHASE_B
C804
0.01μ
C805
0.01μ
1022
GND

Fig. 4

HAND-HELD ELECTRONIC DEVICE WITH A KEYBOARD OPTIMIZED FOR USE WITH THE THUMBS

CROSS-REFERENCE TO RELATED APPLICATIONS

This application is a continuation of U.S. patent application Ser. No. 09/900,585, filed on Jul. 6, 2001 U.S. Pat. No. 6,452,588, which is a continuation of U.S. patent application Ser. No. 09/106,585 U.S. Pat. No. 6,278,442, filed on Jun. 29, 1998, which is a continuation-in-part of U.S. Design application Ser. No. 29/089,942 U.S. Pat. No. D,416,256, entitled Hand-held Messaging Device with Keyboard, filed on Jun. 26, 1998 and assigned to the assignee of the present invention.

BACKGROUND OF THE INVENTION

The present invention is directed toward the field of small, hand-held electronic devices such as personal data assistants (PDA's), personal information managers (PIM's), two-way pagers and the like. In particular, the system and method of the present invention provide the user of the hand-held device with the ability to input data with a minimal amount of key strokes and optimized for use substantially with the thumbs.

In a two-way paging system that provides two-way, full text messaging, there is a need to permit the user to initiate messages and to respond to messages in a timely fashion and with text entirely created by the user. In order to keep the form factor of the two-way pager small enough to be worn on the body of the user, such as with a belt clip, the input device needs to be small, have a minimal number of keys and optimized for use with a minimal number of key strokes. Prior art systems have attempted to address these needs by incorporating virtual keyboards or pen-based input systems for user inputs to the device, but such systems require the user to input data in an unfamiliar manner. Additionally, in a small hand-held messaging device, such as a two-way pager these systems prove awkward to use.

In order to provide a hand-held electronic device that permits a user the opportunity to enter data into an address book, a calendar, a task list, an email message or a similar text file that requires user-generated data, the instant invention is directed to an input device that is oriented to be used substantially through use of the thumbs. This is accomplished first by providing a keyboard with a minimal number of keys, but with the keys representing the alphabet generally placed in the same order as they would appear on a standard keyboard, such as in a standard QWERTY or a DVORAK keyboard layout. The use of a keyboard layout that is familiar to the user enables the user to immediately use the device without having to hunt for the keys he or she wishes to use.

Although the layout is similar to a standard keyboard, the keys are placed at an orientation and in a particular shape that attempts to maximize the surface area of the thumb hitting the key and to provide the user with a comfortable position of the hands for data input. Also, the orientation encourages input by the thumbs, which the inventors of the instant invention have discovered to be faster and more accurate in small hand-held electronic devices than touch-typing or "hunting and pecking" typing.

An additional feature of the invention is thus use of an additional input means for control of functions that might otherwise be controlled by a keyboard that included function keys. To encourage data entry using thumbs and again to

minimize the number of keys on the keyboard, the instant invention also includes a thumb-wheel for control of menus for section of forms and functions relevant to data input. The thumb-wheel is positioned in close proximity to the keyboard to enable the easily transition from thumb-based typing to thumb control of forms and functions.

In addition to hardware features that encourage optimal data entry through the use of thumbs, there are several software features that are designed to minimize keystrokes and aid in entry of data.

The features of this invention, both individually and collectively, have not, to the knowledge of the inventors, been applied to a small hand-held electronic device that requires user-generated data entry. To permit efficient operation of such devices while keeping the form factor of the device small enough to be worn on the body, there is a general need for a hand-held electronic device that can fit in the palm of the hand and that can be operated substantially with the thumbs.

There is a further need for a keyboard for a palm-size data entry device with keys placed at an angle to optimize operation of the keyboard by the use of the thumbs.

There remains another need for a keyboard with keys that are shaped and sized to maximize contact with the thumbs while minimizing the keyboard area required for such keys.

There also remains a need for an auxiliary input device that is to be operated by the thumb for data inputs forms and function control and that, in conjunction with the keyboard, encourages and permits data entry and management through input performed substantially by the thumbs.

There remains still another need for a software-implemented user interface system that is designed, at least in part, to support and encourage data entry through use of the thumbs.

SUMMARY

The present invention overcomes the problems noted above and satisfies the needs in this field for a hand-held electronic device with a keyboard optimized for use with the thumbs. In the preferred embodiment of the present invention, the hand-held electronic device is a two-way paging device that permits full-text, two-way messaging such as email messaging and that includes standard PDA or PIM features such as an address book, an electronic calendar, a task list and other text-based features. These features require user input of text strings that can be lengthy and that cannot be reduced to pre-determined or "canned" strings. Thus, for such a device, the efficient entry of data in a device meant to fit into the palm of one's hand requires that two goals are achieved. First, the data entry must be relatively easy from a user perspective. This means that the user must be somewhat familiar with analogous forms of data entry and not have to be trained to use the data entry for the hand-held device. Second, the form factor does not permit a large number of keys or keys that are very large. Thus efficient use of the keyboard space is required and functions that might be able to be performed by a standard keyboard are off-loaded to an auxiliary input device or are performed, through a minimal number of keystrokes that encourage the use of thumb-based data entry.

To accomplish these goals the invention first optimizes the placement of the keys on the device keyboard. In order to work within the limited space available for the keyboard, it was determined that it was preferable to use keys that were oval or oblong and that were placed at angles designed to facilitate use by thumb typing. An angle for the keys on the right side of the keyboard and a complementary angle for the keys on the left side of the keyboard are chosen based upon observation of the angle at which a user will orient his or her thumbs while thumb-typing.

The invention also minimizes the number of keys available for data input. In the preferred embodiment, only keys for the 26 letters of the English alphabet are available as well as a backspace key, a line feed key, an "alt" key, a "cap" key and a space bar. The alt key enables the user in conjunction the other keys to input numbers and symbols to perform certain functions. The placement of the keys is designed to enhance the user experience while typing with the thumbs by meeting two seemingly opposite goals—minimizing the keyboard footprint while maximizing the likelihood that proper keys will be struck by the thumb-typing user.

The invention also provides additional incentive for the user to use thumb input by providing an input device adjacent to the keyboard, but integral to the overall handheld device. Although other devices can be used in an auxiliary fashion, the preferred device is a thumbwheel that registers movement of the wheel by measuring the number of indents traversed while rolling the wheel and that also registers as an input the depression or "clicking" of the wheel, which is performed by pressing the wheel toward the back of the pager. This clicking of the wheel is similar to the clicking of a mouse associated with a PC or any other input device that registers the depression of a button. The thumbwheel in the preferred embodiment is placed vertically on the two-way paging device so that the user can easily move his or her thumb from the thumbwheel to the keyboard and back for performing functions and retrieving data forms, such as an e-mail template or address book entry template for data entry.

Additionally, various software techniques can be implemented to enhance the thumbtyping use's experience in using the device of the instant invention. In the preferred embodiment, for example, the user can change the capitalization of a particular letter simply by keeping a key depressed for a particular length of time without an intermittent release being detected by the keyboard controller.

The primary advantage of the present invention is that it enables efficient and user-friendly data entry into a palm-sized electronic device by maximizing the potential for user data entry through thumb typing.

These are just a few of the many advantages of the present invention, as described in more detail below. As will be appreciated, the invention is capable of other and different embodiments and its several details are capable of modifications in various respects, all without departing from the spirit of the invention. Accordingly, the drawings and description of the preferred embodiment set forth below are to be regarded as illustrative in nature and not restrictive.

BRIEF DESCRIPTION OF THE DRAWINGS

The present invention satisfies the needs noted above as will become apparent from the following description when read in conjunction with the accompanying drawings wherein:

FIG. 1 is a block diagram of a two-way, full-text, messaging device incorporating a keyboard and an auxiliary data entry device.

FIG. 2 is a frontal view of the hand-held device showing the shape and placement of the keys on the keyboard and the auxiliary input device.

FIG. 3 is a diagram of showing the shape size and placement of the keys on the keyboard.

FIG. 4 is a diagram of the control circuitry for the thumbwheel.

DETAILED DESCRIPTION

Referring now to the drawings, FIG. 1 is a block diagram of the major subsystems and elements comprising a palm-sized, mobile, two-way messaging device that preferably incorporates the invention. In its broadest terms, the messaging device includes a transmitter/receiver subsystem **100** connected to a DSP **200** for digital signal processing of the incoming and outgoing data transmissions, power supply and management subsystem **300**, which supplies and manages power to the overall messaging device components, microprocessor **400**, which is preferably an X86 architecture processor, that controls the operation of the messaging device, display **500**, which is preferably a full graphic LCD, FLASH memory **600**, RAM **700**, serial output and port **800**, keyboard **900** , thumbwheel **1000** and thumbwheel control logic **1010**. In its intended use, a message comes via a wireless data network, such as the Mobitex network, into subsystem **100**, where it is demodulated via DSP **200** and decoded and presented to microprocessor **300** for display on display **500**. To access the display of the message, the user may choose from functions listed under a menu presented as a result of user interaction with thumbwheel **1000**. If the message is an email message, the user may chose to respond to the email by selecting "Reply" from a menu presented on the display through interaction via thumbwheel **1000** or via menu selection from keyboard **900**. In typing the reply, the user can use keyboard **900** to type full text message replies, or insert pre-determined or "canned" response by using either a particular keystroke pattern or through pulling down pre-determined text strings from a menu of items presented on display **500** through the use of thumbwheel **1000**. When the reply to the message is composed the user can initiate the sending of the message preferably by interaction through thumbwheel **1000**, or alternatively, with less efficiency, through a combination of keyboard **900** keystrokes. When the microprocessor **300** receives an indication that the message is to be sent, it processes the message for transport and, by directing and communicating with transmitter/receiver subsystem **100**, enables the reply message to be sent via the wireless communications data network to the intended recipient. Similar interaction through I/O devices keyboard **900** and thumbwheel **1000** can be used to initiate full-text messages or to forward messages to another party. Also, the keyboard **900** and thumbwheel **1000** can be used to permit data entry to an address book resident on the messaging device, or an electronic calendar or log book, or any other function on the messaging device requiring data entry. Preferably, the thumbwheel is a thumbwheel with a push button SPST with quadrature signal outputs, such as that manufactured by Matsushita Electronic Components Co. Ltd. as part number EVQWK2001.

FIG. 2 is a front view of messaging device **10** that incorporates the invention. Shown in FIG. 2 are a plurality of letter keys **901**, and specialized keys **902**, **903**, **904** and **905** and space bar **906**. Also shown is thumbwheel **1000** in its vertical orientation and in association with display **500** and keyboard **900**. In the preferred embodiment, **902** is the alt key, **903** is the cap key. **904** is the line feed key and **905** is the backspace key.

FIG. 3 is a view of a subset of the letter keys **901**, showing the dimensions and relative position of the keys. Shown also is the point **950** that marks the center of keyboard **900**, key dimensions **970**, **971**, **972** and **973**, as well as angle **960** and the rho value **965**, representing curvature of a letter key **901**.

In investigating optimal key placement on the keyboard, it was determined that the keys should be placed at an angle **960** relative to vertical that facilitated easy typing using thumbs. That angle is preferably positive 40 degrees relative to vertical for keys on the right side of the keyboard (where **950** is the center of the keyboard) and negative 40 degrees for the keys on the left side of the keyboard, although complementary angles ranging from 20 degrees to 70 degrees could also be used to accomplish the goal, albeit less optimally, of facilitating thumb typing. Also as shown on FIGS. **2** and **3**, the keys are dispersed across keyboard **900** evenly so that there is sufficient space between the keys to decrease the opportunity for multiple keys being depressed while thumb typing. Additionally, the keys are sized appropriate given the footprint of the messaging device and the keyboard **900**. In its preferred embodiment, the messaging device **10** measures across its face 64 mm by 89 mm, which does not leave much room for keyboard **900** and display **500**. In the preferred embodiment, keyboard **900** occupies over half of the face of the messaging device **10**.

The key shape and dimensions are also key components of the invention. In order to maximize the surface area of the key that a thumb would hit, the keys are preferably oval, and have a rho **965** defining the curvature of the key of 0.4 14, although values may range higher or lower. Other rho values will lead to an acceptable, but not as optimal or aesthetically pleasing shape of keys **901**. As to the key dimensions, the width **970** of the key **901** is 4.8 millimeters (**971** representing the radius of half that value, 2.4 mm) and the length (or height) **972** of the key **901** is 7 millimeters millimeters (**973** representing the radius of half that value, 3.5 mm).

Turning to one of the software features that aids in the device **10** being optimally used for thumb typing is a capitalization feature implemented via software. If a user depresses a key **901**, the operating system detects a key down event. If the key is released after a period of time, the operating system detects a key up event. If upon a key down event, a period of time elapses before a key up event is detected, the operating system determines that a key repeat event has occurred representing a situation where a user has continued to depress a key without releasing it. A key repeat event is then treated by application software residing in either FLASH **600** or RAM **700** as an event that requires the capitalization of the key previously depressed. This feature disables a key repeat feature and substitutes instead a capitalization feature based upon a key repeat. The timing of the key scanning to determine whether a key has been released can be set to permit a slower keyboard response or a faster keyboard response, depending upon user experience or preferences. Although the capitalization function preferably works only to change the state of a letter to a capital, it alternatively could operate to change a capital letter to a lower case letter. The actual display is changed by the application program substituting the value of the capital letter in the register that holds the value of the letter to be displayed. As alternatively implemented, the continued depressing without release of a letter key could result in a key oscillating between upper case and lower case, depending on the length of time the key is depressed.

FIG. **4** is the logic circuitry **1010** associated with thumbwheel **1000**. Thumbwheel **1000** outputs quadrature signals phase A **1021** and phase B **1022**, which are processed by D flip-flops **1031** and **1032** to present signals **1041** W_UP and **1042** W_DN to microprocessor **300**. Signals **1041** and **1042** represent, respectively, a user rolling the thumbwheel up and rolling the thumbwheel down.

Having described in detail the preferred embodiments of the present invention, including the preferred methods of

operation, it is to be understood that this operation could be carried out with different elements and steps. This preferred embodiment is presented only by way of example and is not meant to limit the scope of the present invention which is defined by the following claims.

We claim:

1. A hand-held messaging device, comprising:

a keyboard that is horizontally positioned symmetrically between a left edge and a right edge of a face of the hand-held messaging device and having a plurality of keys arranged in a plurality of rows across the face, wherein each row of keys is arranged in a concave pattern;

a display that is vertically positioned between the keyboard and a top edge of the face and horizontally positioned symmetrically between the left edge and the right edge of the face;

an auxiliary input device; and

a processor coupled to the keyboard, the auxiliary input device and the display that controls the operation of the hand-held messaging device.

2. The hand-held messaging device of claim **1**, wherein each row of keys is arranged in a concave-down pattern.

3. The hand-held messaging device of claim **1**, wherein each row of keys is arranged along an arc.

4. The hand-held messaging device of claim **1**, wherein at least one of the plurality of keys of the keyboard is oblong.

5. The hand-held messaging device of claim **4**, wherein the oblong key is tilted at an angle from a vertical axis extending through a center of the key.

6. The hand-held messaging device of claim **1**, wherein the plurality of keys of the keyboard are oblong.

7. The hand-held messaging device of claim **6**, wherein a first portion of the oblong keys are tilted at a negative angle from vertical and a second portion of the oblong keys are tilted at a positive angle from vertical.

8. The hand-held messaging device of claim **1**, wherein at least one of the plurality of keys of the keyboard is oval.

9. The hand-held messaging device of claim **8**, wherein the oval key is tilted at an angle from a vertical axis extending through a center of the key.

10. The hand-held messaging device of claim **1**, wherein the plurality of keys each have a shape that is contoured for optimal typing with a user's thumbs.

11. The hand-held messaging device of claim **1**, wherein each key of the keyboard is aligned along a vertical axis with a key from an adjacent row.

12. The hand-held messaging device of claim **1**, wherein the keyboard includes three (3) rows of keys, wherein each of the three rows of keys includes a first set of keys that are arranged in a pattern having a positive slope from vertical and a second set of keys that are arranged in a pattern having a negative slope from vertical.

13. The hand-held messaging device of claim **1**, wherein the keyboard includes twenty-six (26) letter keys.

14. The hand-held messaging device of claim **13**, wherein the twenty-six (26) letter keys are arranged in the format of a QWERTY-style keyboard.

15. The hand-held messaging device of claim **14**, further comprising:

a row of functional keys that are horizontally positioned symmetrically or substantially symmetrically between a left edge and a right edge of the face of the hand-held messaging device and vertically positioned between the keyboard and a bottom edge of the hand-held messaging device.

16. The hand-held messaging device of claim **15**, wherein the row of functional keys includes a space bar.

17. The hand-held messaging device of claim **15**, wherein the row of functional keys includes an alt key, and wherein at least one letter key has an associated alternate character that may be input to the processor by simultaneously depressing the letter key and the alt key.

18. The hand-held messaging device of claim **15**, wherein the row of functional keys includes a shift key.

19. The hand-held messaging device of claim **15**, wherein the row of functional keys includes a menu key.

20. The hand-held messaging device of claim **1**, further comprising:

at least one additional functional key positioned above the keyboard.

21. The hand-held messaging device of claim **20**, wherein the additional functional key is a backspace key.

22. The hand-held messaging device of claim **20**, wherein the additional functional key is a home key.

23. The hand-held messaging device of claim **20**, wherein the additional functional key is an escape key.

24. The hand-held messaging device of claim **20**, wherein the additional functional key is a menu key.

25. The hand-held messaging device of claim **20**, wherein the additional functional key is a delete key.

26. The hand-held messaging device of claim **20**, wherein the additional functional key is a cursor-left key.

27. The hand-held messaging device of claim **20**, wherein the additional functional key is a cursor-right key.

28. The hand-held messaging device of claim **1**, wherein the auxiliary input device is a thumb-wheel.

29. The hand-held messaging device of claim **1**, further comprising:

a wireless radio subsystem coupled to the processor that transmits and receives electronic messages from a wireless network; and

a memory device coupled to the processor that stores electronic messages received from the wireless network.

30. The hand-held messaging device of claim **29**, further comprising:

application software executing on the processor, wherein the application software includes an electronic messaging application that receives electronic messages that are wirelessly redirected to the hand-held messaging device from a redirection software application executing on a corporate server.

31. The hand-held messaging device of claim **30**, wherein the application software includes a calendar application.

32. The hand-held messaging device of claim **1**, further comprising:

a rechargeable battery coupled to the processor that supplies power to the hand-held messaging device.

33. A hand-held messaging device, comprising:

a device housing having a face, a bottom surface, and a plurality of connecting surfaces for connecting the face to the bottom surface;

a display mounted within the face of the device housing and horizontally positioned symmetrically between a left edge of the face and a right edge of the face;

a keyboard mounted within the face of the device housing in a position between the display and a bottom edge of the face, wherein the keyboard comprises a QWERTY-style keyboard having a plurality of keys arranged in a plurality of rows across the face; wherein each row of keys is arranged in a concave pattern and is distributed symmetrically across the face of the housing, wherein the keyboard includes a plurality of letter keys and at least one specialized key;

an auxiliary input device mounted within the housing; and

a processor coupled to the keyboard, the auxiliary input device and the display that controls the operation of the hand-held messaging device.

34. The hand-held messaging device of claim **33**, wherein each row of keys is arranged in a concave-down pattern.

35. The hand-held messaging device of claim **33**, wherein each row of keys is arranged along an arc.

36. The hand-held messaging device of claim **33**, wherein the plurality of keys are oblong.

37. The hand-held messaging device of claim **36**, wherein the oblong shaped keys are tilted with respect to a vertical reference through the face of the device housing.

38. The hand-held messaging device of claim **36**, wherein the oblong shaped keys are oval shaped.

39. The hand-held messaging device of claim **33**, wherein the specialized key is a line feed key.

40. The hand-held messaging device of claim **33**, wherein the specialized key is a backspace key.

41. The hand-held messaging device of claim **33**, wherein the keyboard further comprises a row of functional keys.

42. The hand-held messaging device of claim **41**, wherein the row of functional keys includes at least a space bar key, an alt key, and a shift key.

43. The hand-held messaging device of claim **41**, wherein the row of functional keys includes at least a space bar key, a shift key, and a menu key.

44. The hand-held messaging device of claim **33**, wherein the hand-held device is a two-way pager, a personal digital assistant or an electronic organizer.

45. The hand-held messaging device of claim **33**, wherein the auxiliary input device is mounted within one of the connecting surfaces.

46. The hand-held messaging device of claim **33**, wherein the auxiliary input device is a thumbwheel.

47. The hand-held messaging device of claim **33**, wherein the auxiliary input device includes a directional input component for navigating a plurality of menu items presented on the display and a selector switch for selecting a menu item from the plurality of menu items.

48. The hand-held messaging device of claim **33**, further comprising:

a transceiver for transmitting and receiving messages.

49. The hand-held messaging device of claim **48**, further comprising:

a first antenna for receiving messages; and

a second antenna for transmitting messages.

50. The hand-held messaging device of claim **49**, wherein the transceiver further comprises:

a receiver, coupled to the first antenna, for demodulating the received messages; and

a transmitter, coupled to the second antenna, for generating a modulated message.

51. The hand-held messaging device of claim **50**, wherein the transceiver further comprises:

a digital signal processor coupled to the transmitter and the receiver for processing demodulated messages from the receiver, and for providing modulation information to the transmitter.

52. A wireless e-mail device, comprising:

a device housing having a face and a left and right side surface coupled to the face;

a display mounted within the face;

a transceiver for receiving e-mail messages from a wireless network and for transmitting e-mail messages generated on the wireless e-mail device to the wireless network;

a keyboard that is horizontally positioned symmetrically between the left side surface and the right side surface and having a plurality of keys arranged in a plurality of rows across the race, wherein each row of keys is arranged in a concave pattern;

an auxiliary input device mounted within the device housing: and

a processor coupled to the keyboard, the auxiliary input device and the display that controls the operation of the wireless e-mail device.

53. The wireless e-mail device of claim **52**, wherein the auxiliary input device is a thumbwheel.

54. The wireless e-mail device of claim **52**, wherein the auxiliary input device includes a directional input component for navigating a plurality of menu items presented on the display and a selector switch for selecting a menu item form the plurality of menu items.

55. The wireless e-mail device of claim **52**, further comprising an antenna coupled to the transceiver.

56. The wireless e-mail device of claim **52**, further comprising:

a memory for storing an operating system and a plurality of application programs that are executed by the processor to control the operation of the wireless e-mail device.

57. The wireless e-mail device of claim **56**, wherein the operating system assigns a plurality of characters to at least one of the plurality of keys, and wherein the wireless e-mail device includes an auxiliary input device that is used to select one of the plurality of characters by holding down the at least one key and selecting the one of the plurality of characters using the auxiliary input device.

58. The wireless e-mail device of claim **56**, wherein the memory stores a database associating a plurality of first character phrases with a plurality of second character phrases, and wherein the operating system detects one of the plurality of first character phrases input by a user of the wireless e-mail device using the keyboard and substitutes the associated second character phase on the display.

59. The wireless e-mail device of claim **56**, wherein the memory is a flash memory.

60. The wireless e-mail device of claim **56**, further comprising a digital signal processor coupled between the processor and the transceiver.

61. The wireless e-mail device of claim **56**, wherein the plurality of application programs include a messaging application for generating e-mail messages and a calendar application.

62. The wireless e-mail device of claim **61**, wherein the plurality of application programs further include an address book application.

63. The wireless e-mail device of claim **52**, further comprising a serial port for coupling the wireless e-mail device to a host computer.

64. The wireless e-mail device of claim **52**, further comprising a power supply system including a rechargeable battery and an external charger input for receiving a source of electrical charge to recharge the rechargeable battery.

65. The wireless e-mail device of claim **64**, wherein the rechargeable battery is a lithium battery.

66. The wireless e-mail device of claim **64**, wherein the power supply subsystem further includes a voltage regulator coupled to the rechargeable battery for generating a regulated supply voltage for powering the device.

67. The wireless e-mail device of claim **64**, wherein the power supply subsystem further includes connections to a microprocessor for monitoring the operation of the power supply subsystem.

68. A hand-held messaging device, comprising:

a device housing having a face;

a display mounted within the face;

a keyboard mounted within the face of the device housing in a position between the display and a bottom edge of the face, wherein the keyboard comprises a QWERTY-style keyboard having a plurality of keys arranged in a plurality of rows across the face, wherein each row of keys is arranged in a concave pattern and is distributed symmetrically across the face of the housings wherein the keyboard includes a plurality of letter keys and at least one specialized key;

an auxiliary input device mounted within the device housing;

means for receiving e-mail messages from a wireless network and for transmitting e-mail messages generated on the hand-held messaging device to the wireless network.

* * * * *

UNITED STATES PATENT AND TRADEMARK OFFICE

CERTIFICATE OF CORRECTION

PATENT NO.	: 6,611,255 B2	Page 1 of 1
DATED	: August 26, 2003	
INVENTOR(S)	: Griffin et al.	

It is certified that error appears in the above-identified patent and that said Letters Patent is hereby corrected as shown below:

Column 7,
Line 66, "face;" should be -- face, --

Column 10,
Line 36, "housings" should be -- housing, --

Signed and Sealed this

Sixth Day of April, 2004

Jon W. Dudas

JON W. DUDAS
Acting Director of the United States Patent and Trademark Office

EXHIBIT E

US006919879B2

(12) **United States Patent**
Griffin et al.

(10) **Patent No.: US 6,919,879 B2**
(45) **Date of Patent: *Jul. 19, 2005**

(54) **HAND-HELD ELECTRONIC DEVICE WITH A KEYBOARD OPTIMIZED FOR USE WITH THE THUMBS**

(75) Inventors: **Jason T. Griffin**, Waterloo (CA); **David M. Walters**, Waterloo (CA); **John A. Holmes**, Waterloo (CA); **Mihal Lazaridis**, Waterloo (CA)

(73) Assignee: **Research In Motion Limited**, Waterloo (CA)

(*) Notice: This patent issued on a continued prosecution application filed under 37 CFR 1.53(d), and is subject to the twenty year patent term provisions of 35 U.S.C. 154(a)(2).

Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

This patent is subject to a terminal disclaimer.

(21) Appl. No.: **10/205,023**

(22) Filed: **Jul. 25, 2002**

(65) **Prior Publication Data**

US 2003/0020692 A1 Jan. 30, 2003

Related U.S. Application Data

(63) Continuation-in-part of application No. 09/663,972, filed on Sep. 19, 2000, which is a continuation-in-part of application No. 09/106,585, filed on Jun. 29, 1998, now Pat. No. 6,278,442, which is a continuation-in-part of application No. 29/089,942, filed on Jun. 26, 1998, now Pat. No. Des. 416,256.

(60) Provisional application No. 60/307,755, filed on Jul. 25, 2001.

(51) **Int. Cl.**^{7} **G09G 5/00**

(52) **U.S. Cl.** .... **345/168**; 341/20; 341/21; 341/22; 200/5 R; 200/302.1; 200/302.2; 400/486; 400/489

(58) **Field of Search** .... 345/156, 168–172, 345/184, 102; 341/20–22; 200/5 R, 302.1–302.2, 1 TK, 11 TW; 400/486, 489; D14/247, 252

(56) **References Cited**

U.S. PATENT DOCUMENTS

4,029,915 A	6/1977	Ojima
4,449,839 A	5/1984	Bleuer

(Continued)

FOREIGN PATENT DOCUMENTS

EP	0267801	5/1988
EP	0278169	8/1988

(Continued)

OTHER PUBLICATIONS

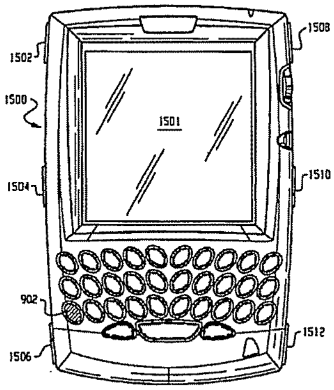
Anonymous: Triangular Toggle Keys for Touch–Tone Phones; IBM Technical Disclosure Bulletin, vol. 31, No. 1, Jun. 1, 1988, pp. 47–49, New York, US.

Primary Examiner—Henry N. Tran
(74) *Attorney, Agent, or Firm*—Jones Day; Krishna K. Pathiyal; Robert C. Liang

(57) **ABSTRACT**

A hand-held electronic device with a keyboard optimized for use with the thumbs is provided. The handheld device includes a keyboard, a display, and a processor. The keyboard is horizontally positioned symmetrically between a left edge and a right edge of a face of the hand-held messaging device. The keyboard has a plurality of keys arranged in a plurality of rows across the face, wherein each row of keys is arranged in a concave pattern. The display is vertically positioned between the keyboard and a top edge of the face and is horizontally positioned symmetrically between the left edge and the right edge of the face. The processor is coupled to the keyboard and the display, and controls the operation of the hand-held messaging device.

128 Claims, 14 Drawing Sheets



U.S. PATENT DOCUMENTS

D278,341	S		4/1985	Scheid	
D293,241	S		12/1987	Wan et al.	
D312,628	S		12/1990	Yokoi et al.	
D313,401	S		1/1991	Tanabe	
D313,413	S		1/1991	Langton	
5,059,048	A	*	10/1991	Sirkin	400/486
5,184,830	A		2/1993	Okada et al.	
5,217,295	A		6/1993	Tortola et al.	
5,288,158	A		2/1994	Matias	
5,336,001	A	*	8/1994	Lichtenberg	400/489
5,337,346	A		8/1994	Uchikura	
5,360,280	A	*	11/1994	Camacho et al.	400/489
5,367,298	A		11/1994	Axthelm	
D357,253	S		4/1995	Wong	
5,410,141	A		4/1995	Koenck et al.	
5,410,333	A		4/1995	Conway	
5,426,449	A		6/1995	Danziger	
D359,920	S		7/1995	Sakamoto	
5,436,954	A		7/1995	Nishiyama et al.	
5,457,454	A		10/1995	Sugano	
D367,043	S		2/1996	Ross et al.	
5,500,643	A	*	3/1996	Grant	341/22
5,543,787	A		8/1996	Karidis et al.	
5,563,631	A		10/1996	Masunaga	
5,575,576	A		11/1996	Roysden, Jr.	
5,600,790	A		2/1997	Barnstijn et al.	
5,606,712	A		2/1997	Hidaka	
5,611,031	A		3/1997	Hertzfeld et al.	
D381,021	S	*	7/1997	Williams et al.	
5,659,307	A		8/1997	Karidis et al.	
5,661,605	A		8/1997	Conway	
D383,756	S		9/1997	Henderson et al.	
5,672,108	A		9/1997	Lam et al.	
D386,497	S	*	11/1997	Huslig et al.	D14/191
5,689,253	A	*	11/1997	Hargreaves et al.	341/22
D390,509	S		2/1998	Antzinas et al.	
5,737,394	A		4/1998	Anderson et al.	
5,786,776	A		7/1998	Kisaichi et al.	
D397,369	S		8/1998	Rissman	
D397,728	S		9/1998	Yuen et al.	
D399,537	S		10/1998	Chi et al.	
5,818,437	A		10/1998	Grover et al.	
5,825,353	A		10/1998	Will	
5,827,082	A		10/1998	Laine	
D402,572	S		12/1998	Han	
D403,362	S		12/1998	Fai	
5,861,821	A		1/1999	Kato et al.	
5,893,798	A		4/1999	Stambolic et al.	
5,915,228	A		6/1999	Kunihiro et al.	
5,920,308	A		7/1999	Kim	
5,931,873	A		8/1999	Cisar	
5,963,197	A		10/1999	Bacon et al.	
5,974,238	A		10/1999	Chase, Jr.	
D416,256	S		11/1999	Griffin et al.	
5,982,520	A		11/1999	Weiser et al.	
5,995,026	A		11/1999	Sellers	
6,005,496	A		12/1999	Hargreaves et al.	
6,006,351	A		12/1999	Peretz et al.	
6,009,333	A		12/1999	Chaco	
6,014,429	A		1/2000	LaPorta et al.	
6,014,573	A		1/2000	Lehtonen et al.	
D420,351	S		2/2000	Waldner	
6,023,779	A		2/2000	Fullam et al.	
6,047,047	A		4/2000	Aldridge et al.	
6,047,196	A		4/2000	Makela et al.	
6,049,796	A		4/2000	Siitonen et al.	
6,052,070	A		4/2000	Kivela et al.	
6,084,576	A	*	7/2000	Leu et al.	345/168
6,091,956	A		7/2000	Hollenberg	
6,094,197	A		7/2000	Buxton et al.	
6,102,594	A		8/2000	Strom	
6,103,979	A		8/2000	Motoyama et al.	
6,107,997	A		8/2000	Ure	
D432,511	S		10/2000	Eckholm	
D433,017	S		10/2000	Martinez	
D433,460	S		11/2000	Griffin et al.	
6,148,261	A		11/2000	Obradovich et al.	
6,157,323	A		12/2000	Tso et al.	
D436,591	S		1/2001	Abston et al.	
6,212,412	B1		4/2001	Rogers et al.	
D441,733	S		5/2001	Do et al.	
6,243,789	B1		6/2001	Hasbun et al.	
6,278,442	B1		8/2001	Griffin et al.	
6,295,052	B1		9/2001	Kato et al.	
6,297,795	B1		10/2001	Kato et al.	
6,304,261	B1		10/2001	Shields et al.	
6,304,431	B1		10/2001	Kim	
6,310,609	B1	*	10/2001	Morgenthaler	345/170
D451,079	S		11/2001	Ali	
D454,349	S		3/2002	Makidera et al.	
D454,849	S		3/2002	Eckholm	
6,356,258	B1		3/2002	Kato et al.	
6,374,277	B2		4/2002	Vong et al.	
D456,794	S		5/2002	Laverick et al.	
6,385,463	B1		5/2002	Lieberman et al.	
6,396,482	B1		5/2002	Griffin et al.	
D458,239	S		6/2002	Shim et al.	
D459,327	S		6/2002	Ali	
D460,068	S		7/2002	Lanzaro et al.	
D460,493	S		7/2002	Griffin et al.	
D461,803	S		8/2002	Griffin et al.	
6,452,588	B2		9/2002	Griffin et al.	
D464,995	S		10/2002	Griffin et al.	
6,459,968	B1		10/2002	Kochie	
6,489,950	B1		12/2002	Griffin et al.	
6,507,336	B1		1/2003	Lunsford	
D472,225	S		3/2003	Griffin	
6,535,749	B1		3/2003	Iwata et al.	
6,538,651	B1		3/2003	Haymann et al.	
D472,551	S		4/2003	Griffin	
D473,226	S		4/2003	Griffin	
D476,985	S		7/2003	Griffin	
D478,585	S		8/2003	Griffin	
6,611,254	B1		8/2003	Griffin et al.	
D479,233	S		9/2003	Griffin	
D480,722	S		10/2003	Griffin	
6,630,924	B1		10/2003	Peck	
6,647,367	B2		11/2003	McArthur et al.	
2001/0044828	A1		11/2001	Kikinis	

FOREIGN PATENT DOCUMENTS

EP	0538020	4/1993
EP	0685801	12/1995
EP	0732646	9/1996
EP	0760291	3/1997
EP	1143327	10/2001
WO	9833111	7/1998
WO	9937025	7/1999
WO	0030381	5/2000
WO	0038041	6/2000
WO	0074240	12/2000
WO	0150335	7/2001

* cited by examiner

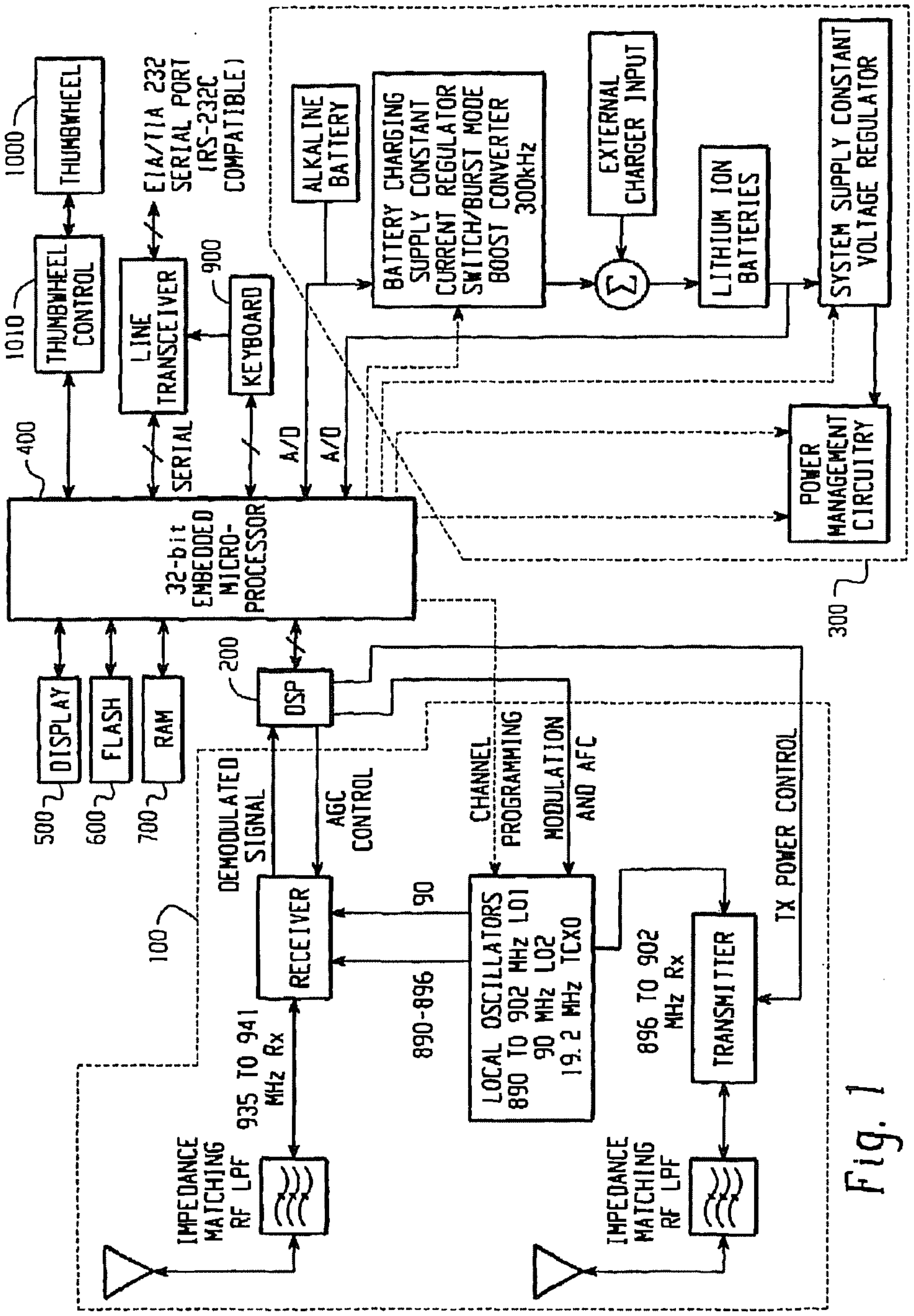

1000
THUMBWHEEL
1010
THUMBWHEEL CONTROL
EIA/TIA 232 SERIAL PORT (RS-232C COMPATIBLE)
LINE TRANSCEIVER
900
KEYBOARD
400
SERIAL
A/D
A/D
32-bit EMBEDDED MICRO-PROCESSOR
ALKALINE BATTERY
BATTERY CHARGING SUPPLY CONSTANT CURRENT REGULATOR SWITCH/BURST MODE BOOST CONVERTER 300kHz
EXTERNAL CHARGER INPUT
Σ
LITHIUM ION BATTERIES
SYSTEM SUPPLY CONSTANT VOLTAGE REGULATOR
POWER MANAGEMENT CIRCUITRY
300
DISPLAY
FLASH
RAM
500
600
700
200
DSP
DEMODULATED SIGNAL
AGC CONTROL
CHANNEL PROGRAMMING
MODULATION AND AFC
TX POWER CONTROL
100
RECEIVER
90
890-896
LOCAL OSCILLATORS 890 TO 902 MHz LO1 90 MHz LO2 19.2 MHz TCXO
896 TO 902 MHz Rx
TRANSMITTER
935 TO 941 MHz Rx
IMPEDANCE MATCHING RF LPF
IMPEDANCE MATCHING RF LPF

Fig. 1

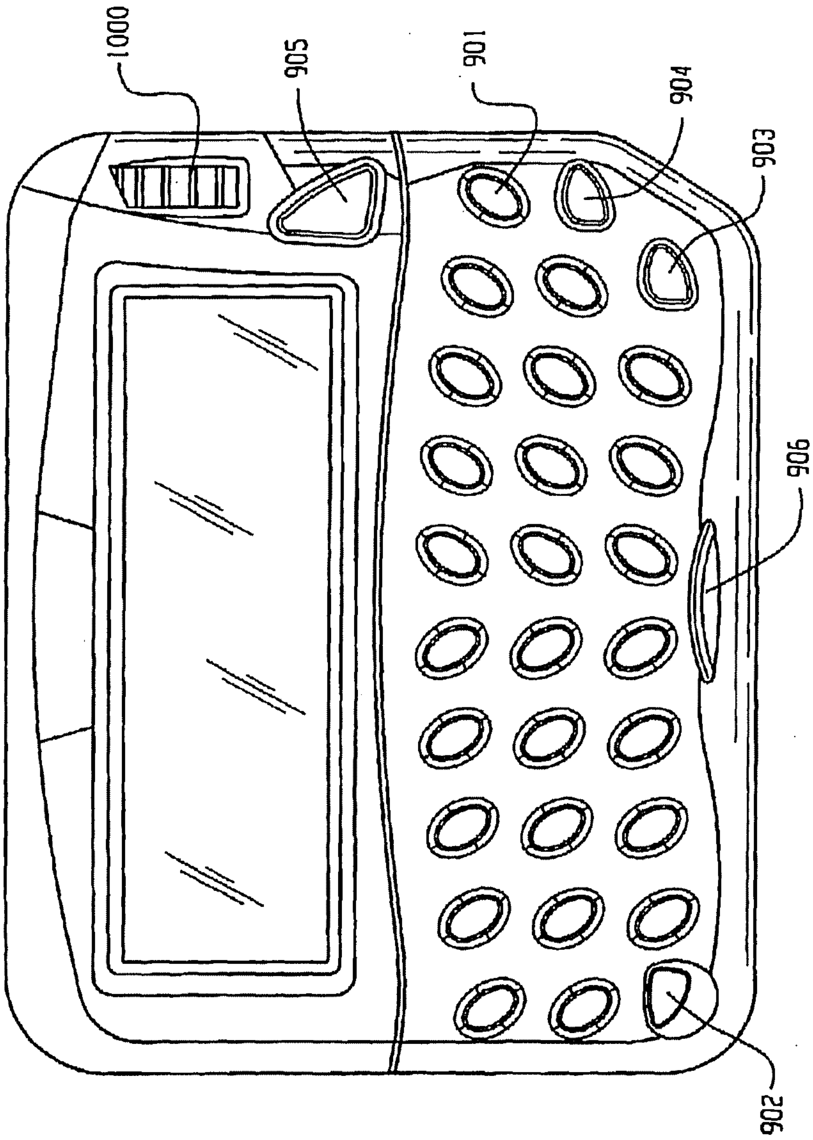


Fig. 2

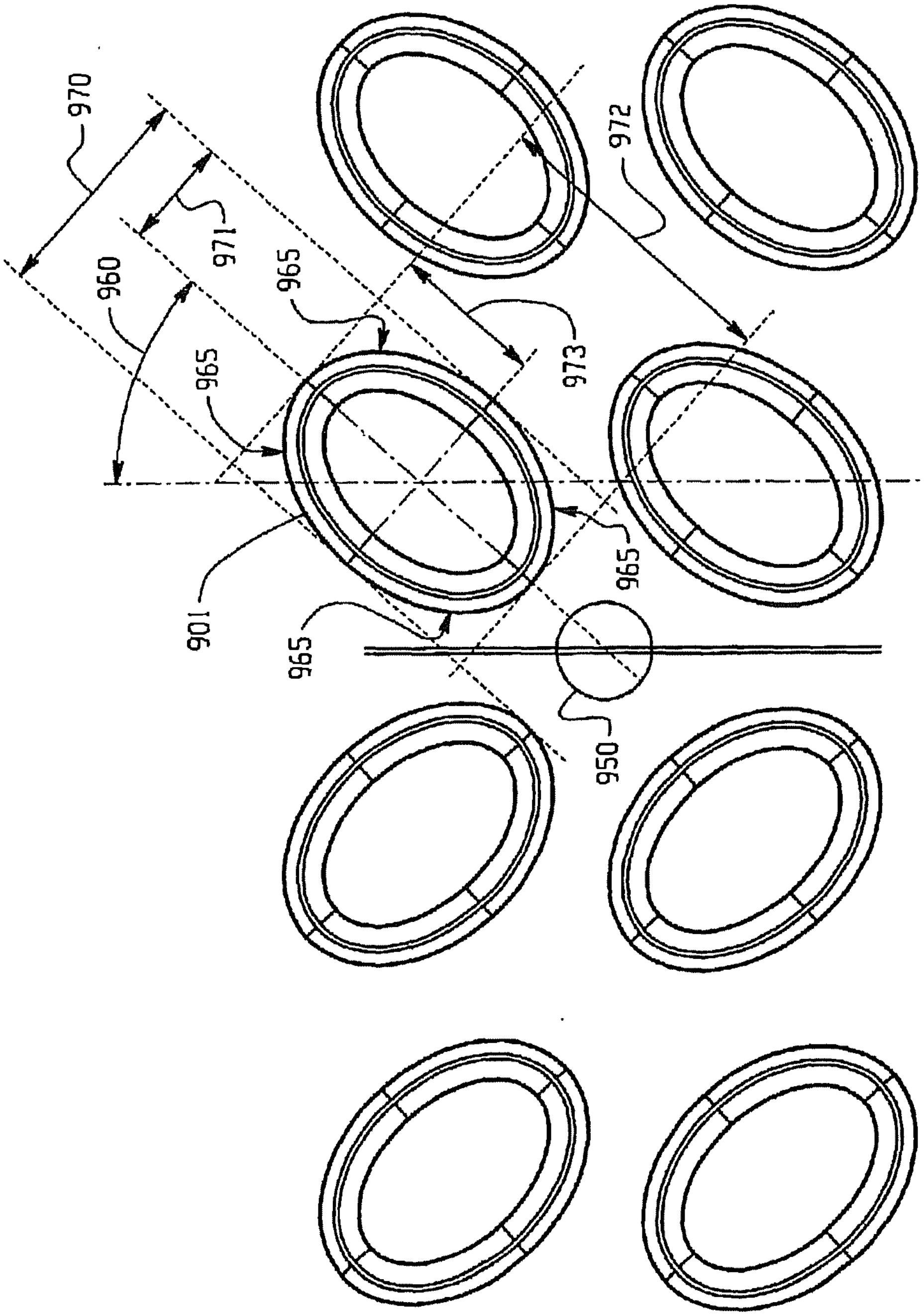


Fig. 3

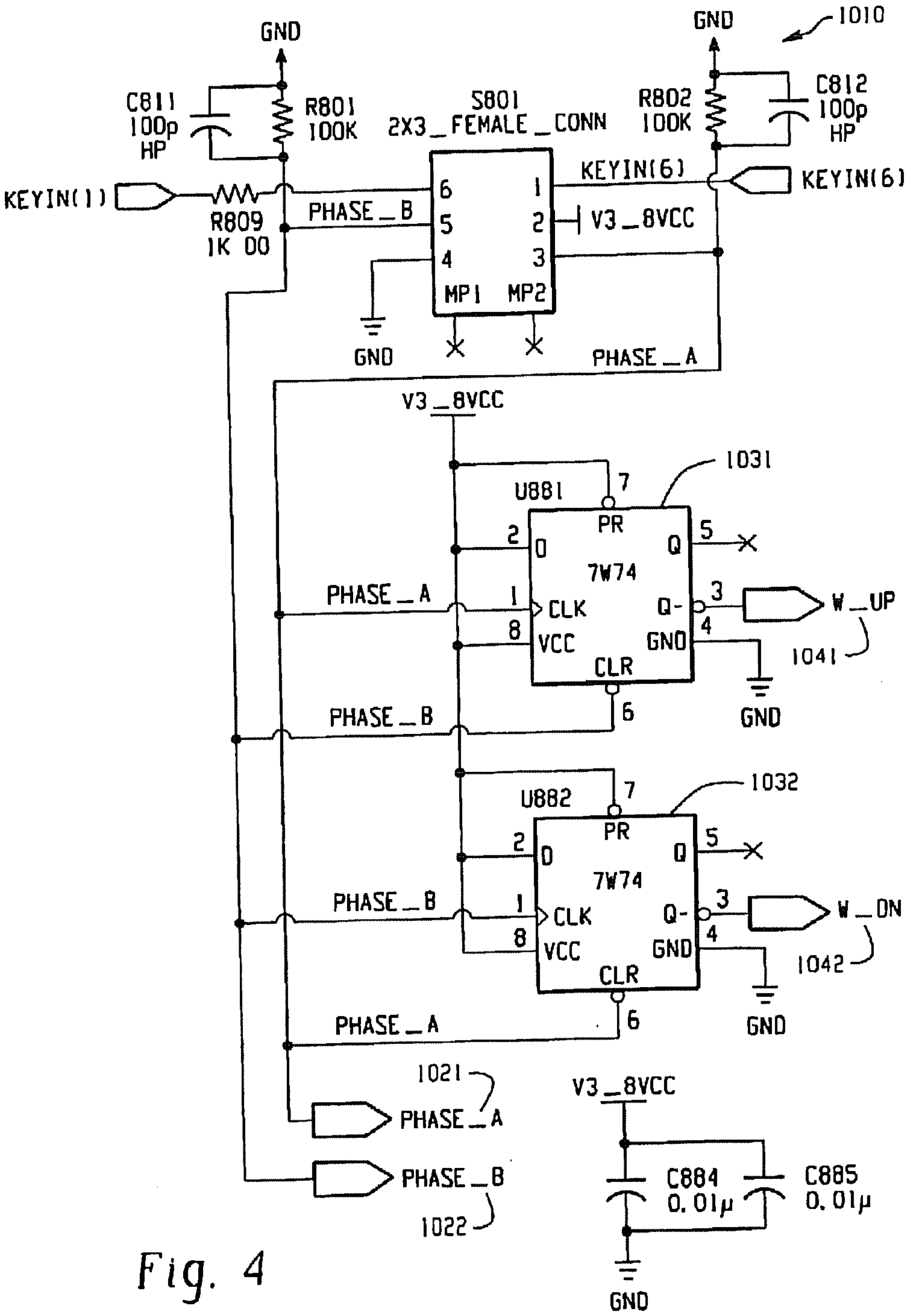

GND
GND
1010
C811
100p
HP
R801
100K
S801
2X3_FEMALE_CONN
R802
100K
C812
100p
HP
KEYIN(1)
R809
1K 00
PHASE_B
6
5
4
1
2
3
MP1
MP2
KEYIN(6)
KEYIN(6)
V3_8VCC
GND
PHASE_A
V3_8VCC
1031
U881
7
2
D
PR
Q
5
7W74
PHASE_A
1
CLK
Q-
3
W_UP
8
VCC
GND
4
1041
CLR
PHASE_B
6
GND
U882
7
1032
2
D
PR
Q
5
7W74
PHASE_B
1
CLK
Q-
3
W_DN
8
VCC
GND
4
1042
CLR
PHASE_A
6
GND
1021
PHASE_A
V3_8VCC
PHASE_B
C884
0.01μ
C885
0.01μ
1022
GND

Fig. 4

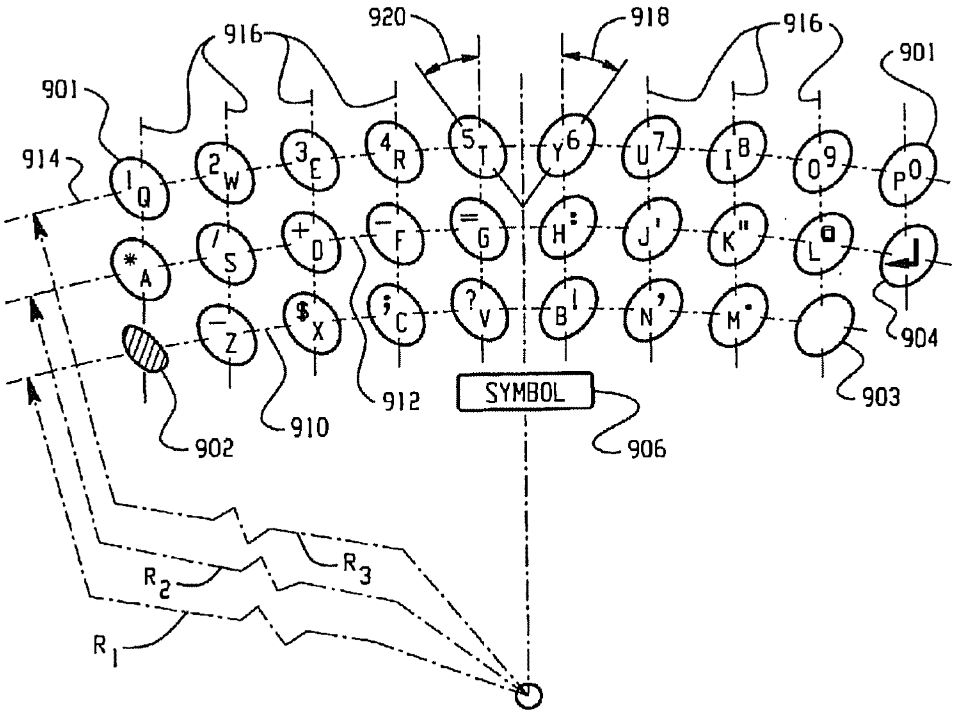

920
918
916
916
901
901
914
1 Q
2 W
3 E
4 R
5 T
Y 6
U 7
I 8
O 9
P 0
* A
/ S
+ D
− F
= G
H :
J '
K "
L @
904
− Z
$ X
; C
? V
B !
N ,
M .
903
902
910
912
SYMBOL
906
R3
R2
R1

Fig. 5

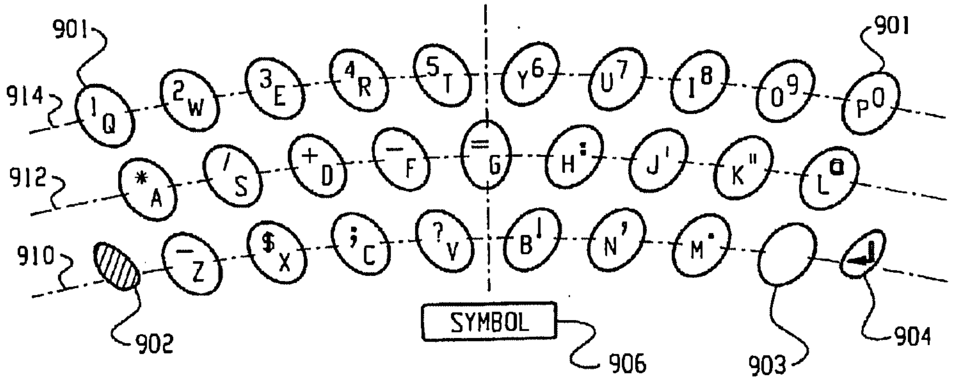

901
901
914
912
910
1 Q
2 W
3 E
4 R
5 T
Y 6
U 7
I 8
O 9
P 0
* A
/ S
+ D
− F
= G
H :
J '
K "
L @
− Z
$ X
; C
? V
B !
N ,
M .
902
SYMBOL
906
903
904

Fig. 6

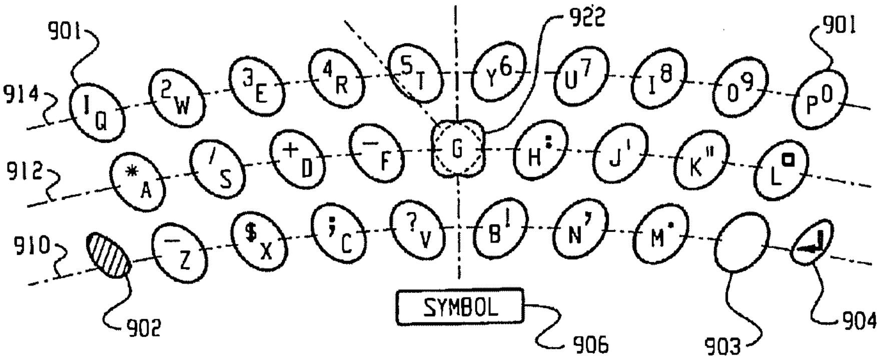

901
922
901
914
Q
W
E
R
T
Y
U
I
O
P
912
A
S
D
F
G
H
J
K
L
910
Z
X
C
V
B
N
M
902
SYMBOL
906
903
904

Fig. 7

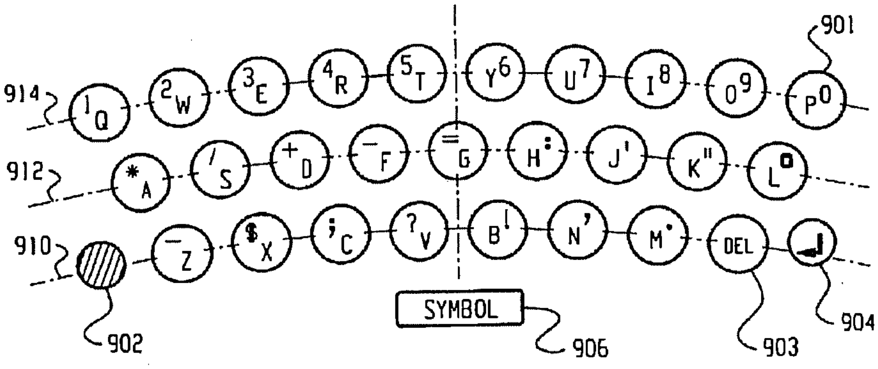

901
914
Q
W
E
R
T
Y
U
I
O
P
912
A
S
D
F
G
H
J
K
L
910
Z
X
C
V
B
N
M
DEL
902
SYMBOL
906
903
904

Fig. 8

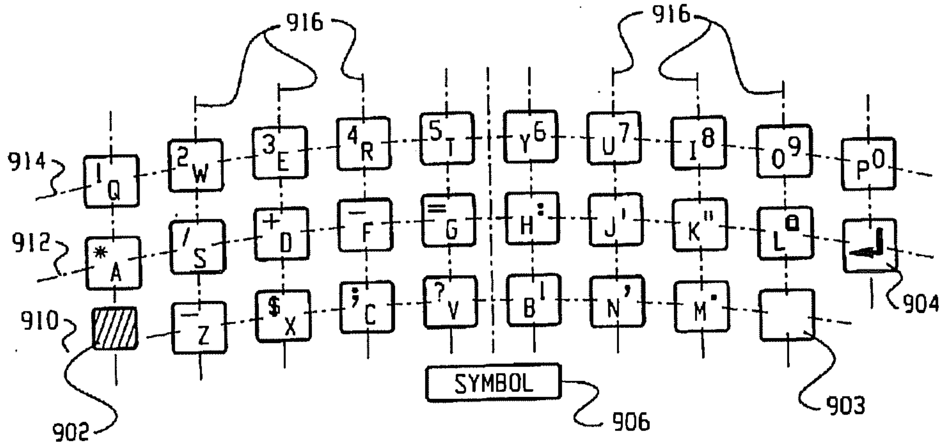

916
916
914
912
910
902
1 Q
2 W
3 E
4 R
5 T
Y 6
U 7
I 8
O 9
P 0
* A
/ S
+ D
— F
= G
H :
J '
K "
L @
904
— Z
$ X
; C
? V
B !
N ,
M .
SYMBOL
906
903

Fig. 9

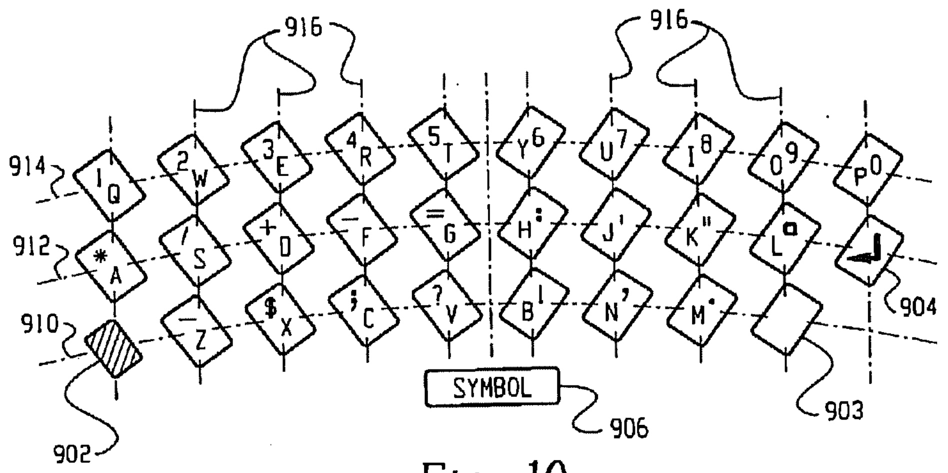

916
916
914
912
910
902
1 Q
2 W
3 E
4 R
5 T
Y 6
U 7
I 8
O 9
P 0
* A
/ S
+ D
— F
= G
H :
J '
K "
L @
904
— Z
$ X
; C
? V
B !
N ,
M .
SYMBOL
906
903

Fig. 10

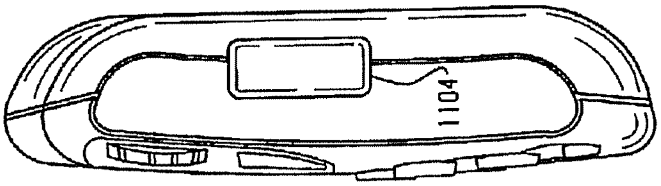


Fig. 11c

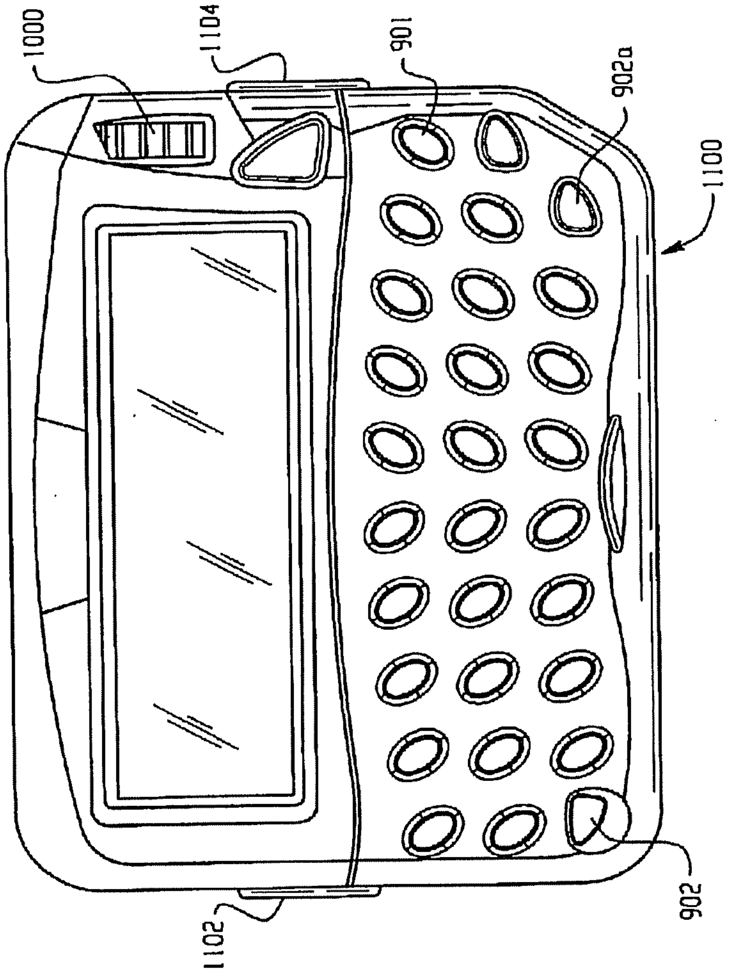


Fig. 11a

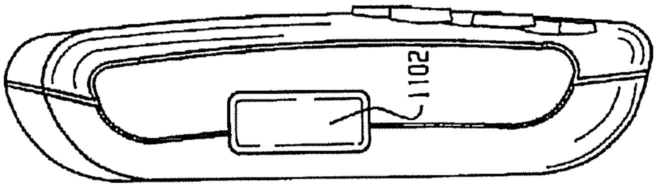


Fig. 11b

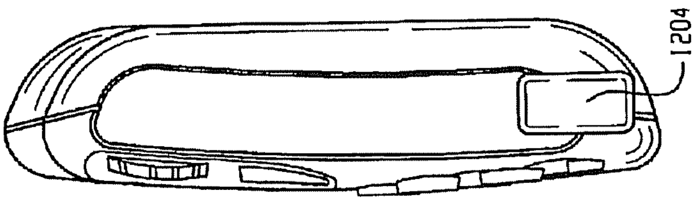


Fig. 12c

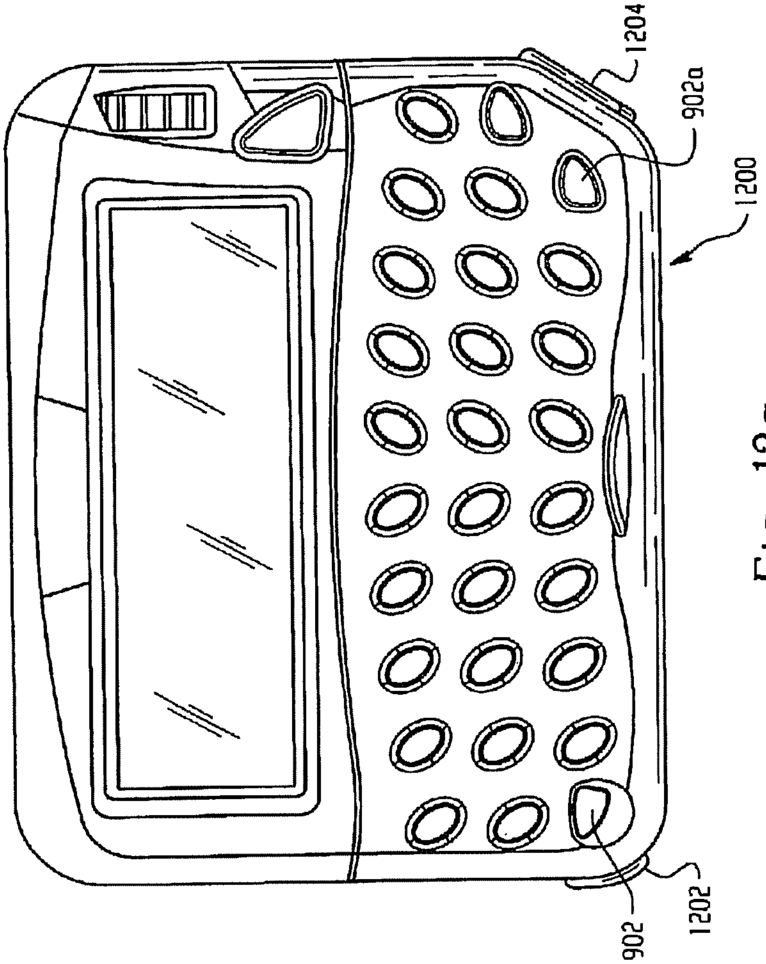


Fig. 12a

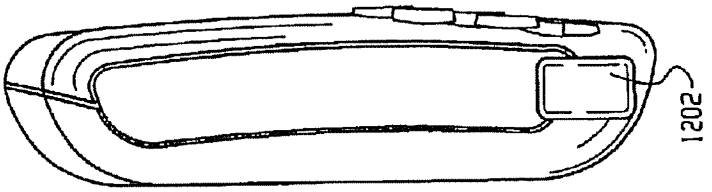


Fig. 12b

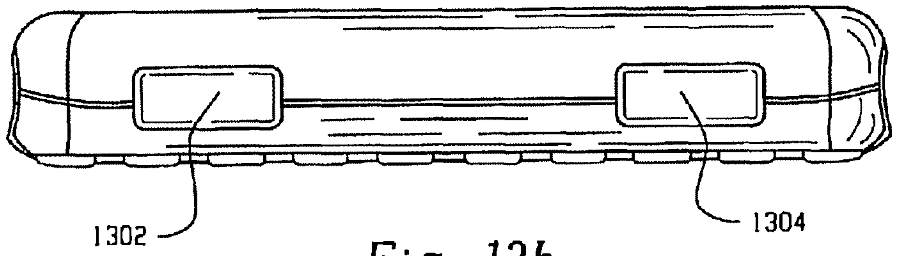


Fig. 13b

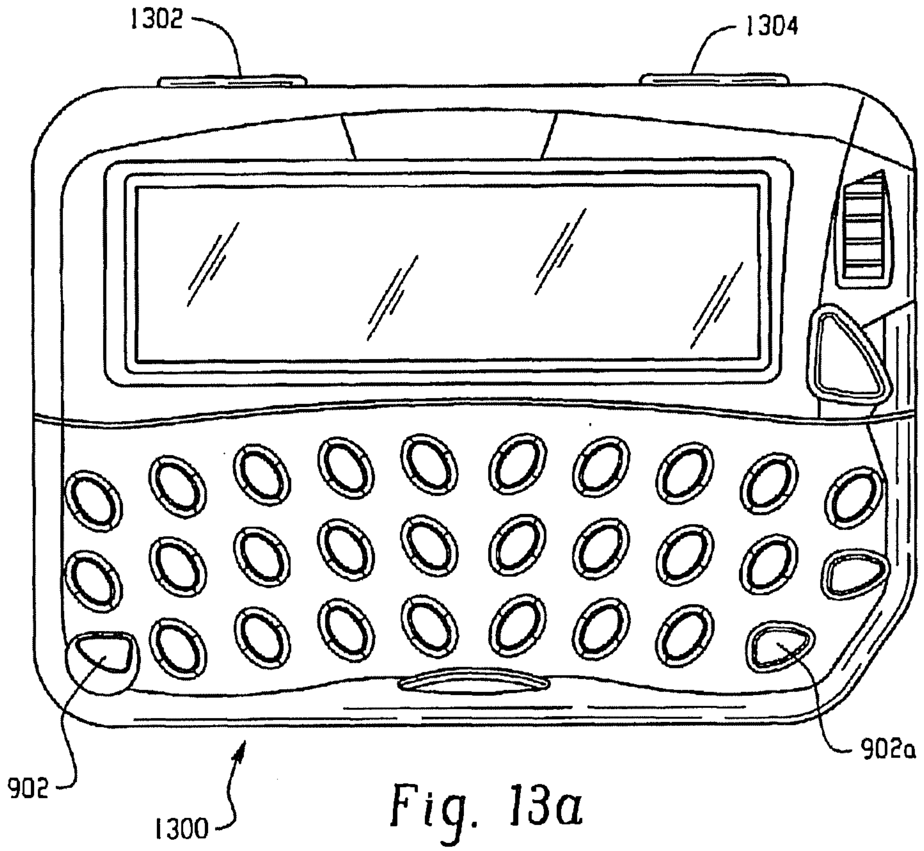


Fig. 13a

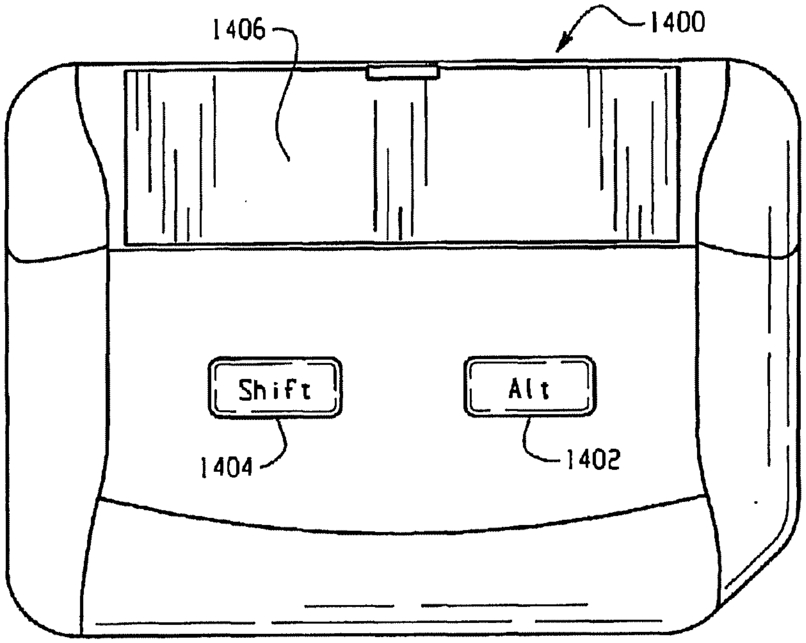


Fig. 14a

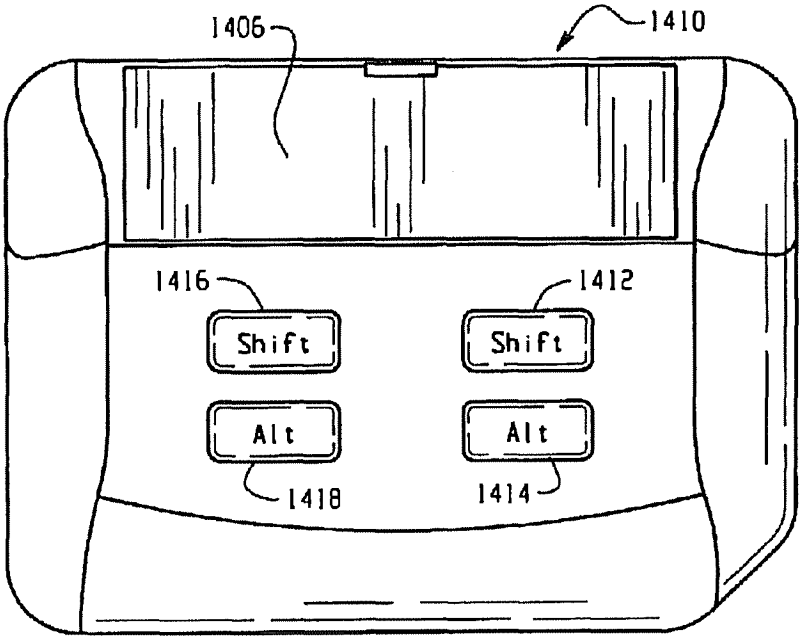


Fig. 14b

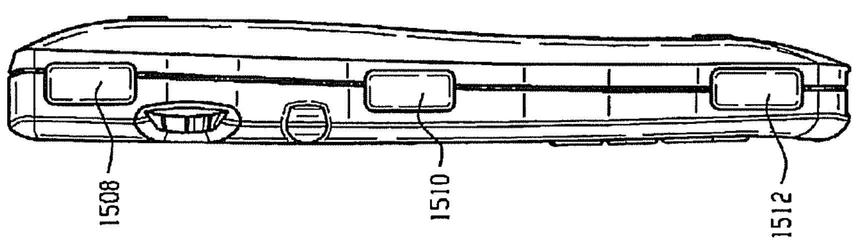


Fig. 15c

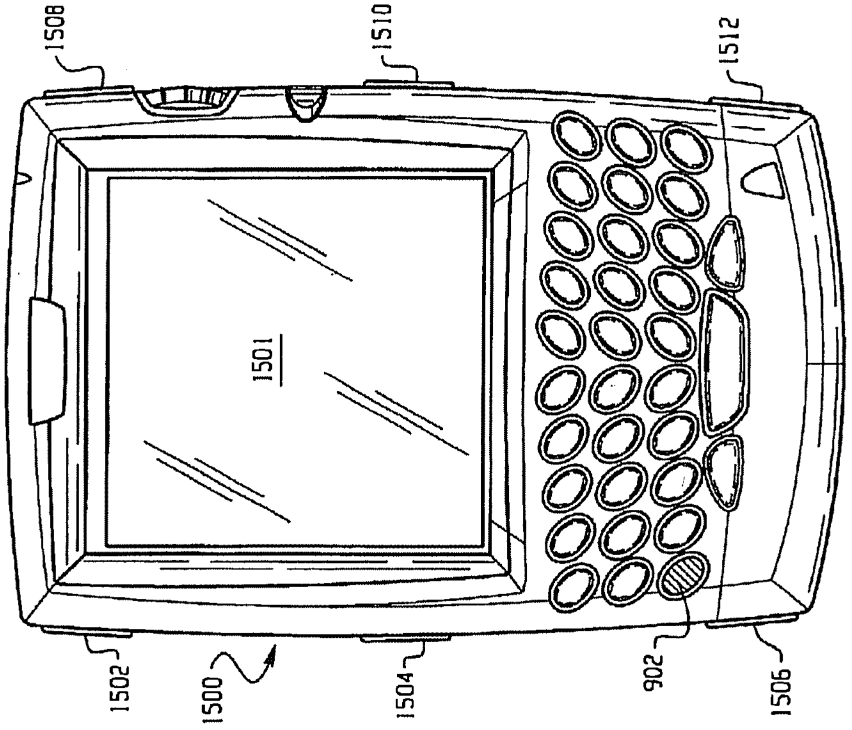


Fig. 15a

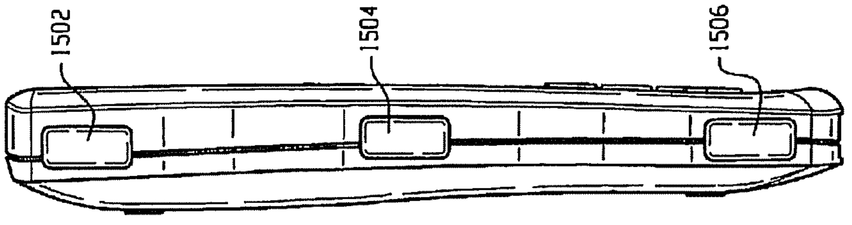


Fig. 15b

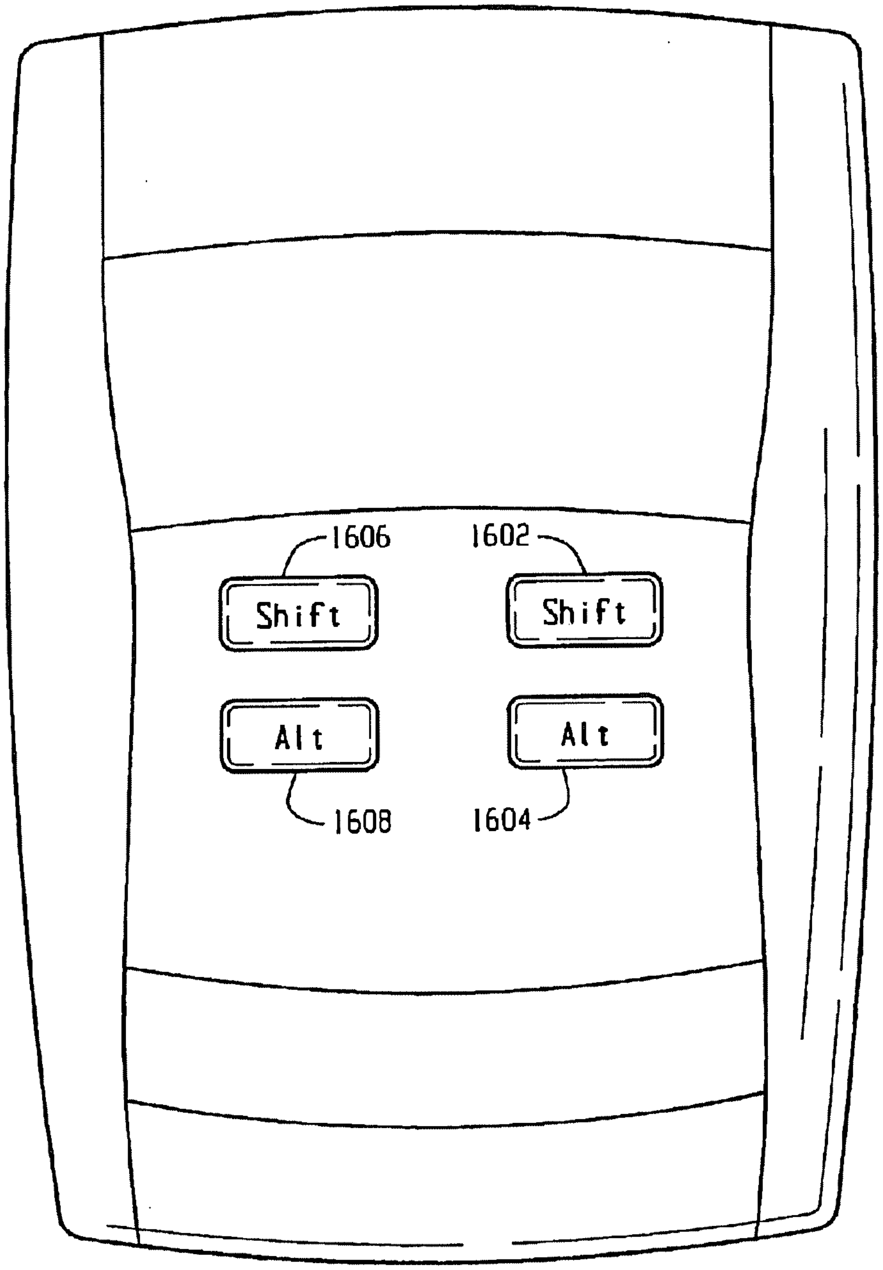


Fig. 16

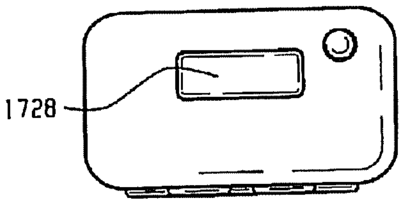


Fig. 17d

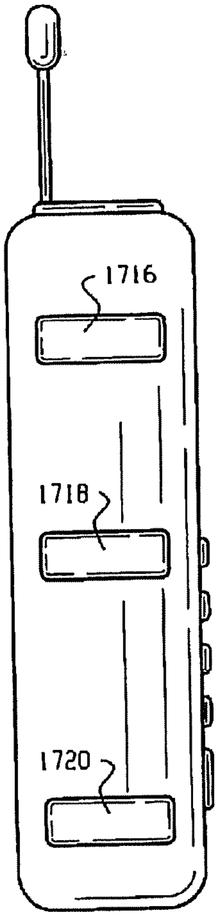


Fig. 17b

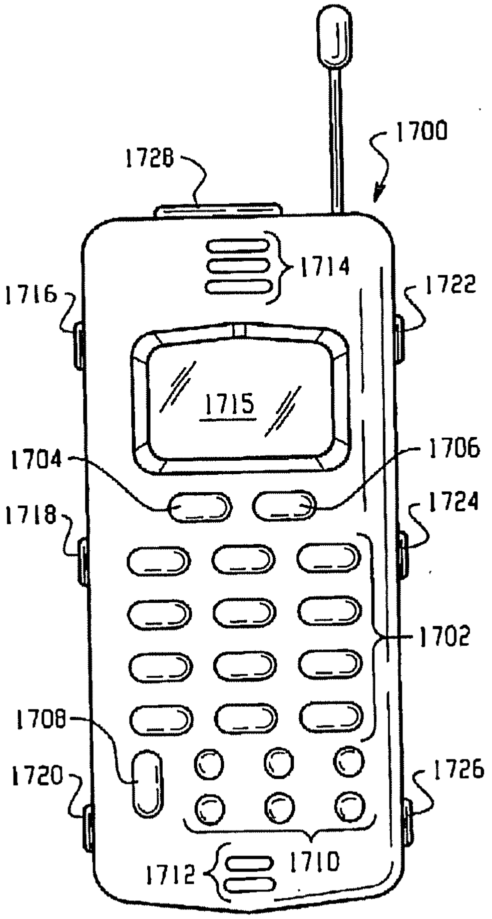


Fig. 17a

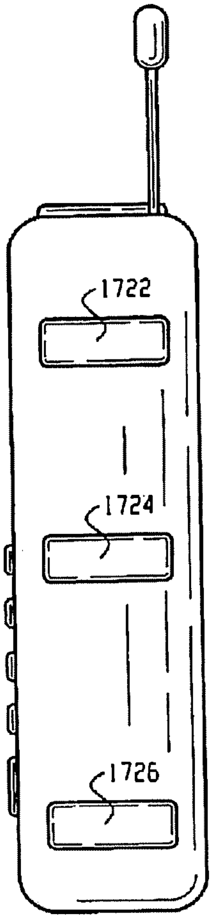


Fig. 17c

HAND-HELD ELECTRONIC DEVICE WITH A KEYBOARD OPTIMIZED FOR USE WITH THE THUMBS

CROSS-REFERENCE TO RELATED APPLICATION

This application claims priority from and is related to the following prior application: "Hand-Held Electronic Device with a Keyboard Optimized for Use with the Thumbs", U.S. Provisional Application Ser. No. 60/307,755, filed on Jul. 25, 2001. In addition, this application is a continuation-in-part of U.S. patent application Ser. No. 09/663,972, filed on Sep. 19, 2000, which is a continuation-in-part of U.S. patent application Ser. No. 09/106,585, filed on Jun. 29, 1998 (now U.S. Pat. No. 6,278,442), which is a continuation-in-part of U.S. Design application Ser. No. 29/089,942, entitled "Hand-held Messaging Device with Keyboard", filed on Jun. 26, 1998 (now U.S. Pat. No. Des. 416,256) and assigned to the assignee of the present invention. These prior applications, including the entire written descriptions and drawing figures, are hereby incorporated into the present application by reference.

BACKGROUND

The arrangements described herein are directed toward the field of small, hand-held electronic devices such as personal data assistants (PDAs), personal information managers (PIMs), two-way pagers and the like. In particular, the described systems and methods provide the user of a hand-held device with the ability to input data with a minimal amount of key strokes, and includes a keyboard structure that is optimized for use substantially with the thumbs.

In a two-way paging system that provides two-way, full text messaging, there is a need to permit the user to initiate messages and to respond to messages in a timely fashion and with text entirely created by the user on a communication device. In order to keep the form factor of the device small enough to be worn on the body of the user, such as with a belt clip, the input device should be small, have a minimal number of keys, and be optimized for use with a minimal number of key strokes. Known systems have attempted to address these needs by incorporating virtual keyboards or pen-based systems for user inputs to the device, but such systems require the user to input data in an unfamiliar manner. Additionally, in a small hand-held messaging device, such as a two-way pager, these systems prove awkward to use.

In order to provide a hand-held electronic device that permits a user the opportunity to enter data into an address book, a calendar, a task list, an email or other message or a similar text file that requires user-generated data, this application describes an input device that is oriented to be operated substantially through use of the thumbs. This is accomplished first by providing a keyboard with a minimal number of keys, but with the keys representing the alphabet generally placed in the same order as they would appear on a standard keyboard, such as in a standard QWERTY or a DVORAK keyboard layout. The use of a keyboard layout that is familiar to the user enables the user to immediately use the device without having to hunt for the keys he or she wishes to use.

Although the layout is similar to a standard keyboard, the keys are placed at an orientation and in a particular shape that attempts to maximize the surface area of the thumb hitting the key and to provide the user with a comfortable position of the hands for data input. Also, the orientation encourages input by the thumbs, which the inventors of the present invention have discovered to be faster and more accurate in small hand-held electronic devices than touch-typing or "hunting and pecking" typing.

The device preferably includes an additional input means for control of functions that might otherwise be controlled by a keyboard that included function keys. To encourage data entry using thumbs and again to minimize the number of keys on the keyboard, the device may also include a thumb-wheel for control of menus to select forms and functions relevant to data input. The thumb-based wheel is preferably position in close proximity to the keyboard to enable the easy transition from thumb-based typing to thumb control of forms and functions via the thumb-wheel.

In addition to hardware features that encourage optimal data entry through the use of thumbs, several software features that are designed to minimize keystrokes and aid data entry are also provided.

SUMMARY

A hand-held electronic device with a keyboard optimized for use with the thumbs is provided. The hand-held device includes a keyboard, a display, and a processor. The keyboard is horizontally positioned symmetrically between a left edge and a right edge of a face of the hand-held messaging device. The keyboard has a plurality of keys arranged in a plurality of rows across the face, wherein each row of keys is arranged in a concave pattern. The display is vertically positioned between the keyboard and a top edge of the face and is horizontally positioned symmetrically between the left edge and the right edge of the face. The processor is coupled to the keyboard and the display, and controls the operation of the hand-held messaging device.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. **1** is a block diagram of the major subsystems and elements comprising a palm-sized, mobile, two-way messaging device that may incorporate a keyboard optimized for use with the thumbs;

FIG. **2** is a front view of an exemplary messaging device having a keyboard that is optimized for use with the thumbs;

FIG. **3** is a view of a subset of the letter keys shown in FIG. **2**, illustrating exemplary dimensions and relative positions of the keys;

FIG. **4** is the logic circuitry associated with the thumb-wheel of FIGS. **1** and **2**;

FIG. **5** is a diagram showing one exemplary embodiment of a keyboard that is optimized for use with the thumbs;

FIG. **6** is a diagram showing another exemplary embodiment of a keyboard optimized for use with the thumbs;

FIG. **7** is another embodiment of the keyboard shown in FIG. **6** having a special center key with a vertically symmetrical key shape and orientation;

FIG. **8** shows a keyboard embodiment with circular keys;

FIG. **9** shows a keyboard embodiment with square keys;

FIG. **10** shows a keyboard embodiment having rectangular keys;

FIGS. **11***a* through **11***c* show front and side views of a hand-held electronic device incorporating an alternative functional key arrangement;

FIGS. **12***a* through **12***c* are diagrams showing front and side views of a hand-held electronic device incorporating another alternative functional key arrangement;

FIGS. **13***a* and **13***b* show front and top views of a hand-held electronic device incorporating a further alternative functional key arrangement;

FIGS. **14***a* and **14***b* show rear views of a hand-held electronic device in which two additional functional key arrangements are implemented;

FIGS. **15***a* through **15***c* show front and side views of another exemplary hand-held electronic device incorporating an alternative functional key arrangement;

FIG. **16** is a rear view of a device, such as shown in FIG. **15**, in which another functional key arrangement is implemented; and

FIGS. **17***a* through **17***d* show front, side and top views of a further exemplary hand-held electronic device incorporating alternative functional key arrangements.

DETAILED DESCRIPTION

Referring now to the drawings, FIG. **1** is a block diagram of the major subsystems and elements comprising a palm-sized, mobile, two-way messaging device that may incorporate a keyboard optimized for use with the thumbs. The exemplary messaging device shown in FIG. **1** includes a wireless radio transmitter/receiver subsystem **100** connected to a DSP **200** for digital signal processing of the incoming and outgoing data transmissions, power supply and management subsystem **300**, which supplies and manages power to the overall messaging device components, microprocessor **400**, which is preferably an X86 architecture processor, which controls the operation of the messaging device, display **500**, which is preferably a full graphic LCD, FLASH memory **600**, RAM **700**, serial port **800**, keyboard **900**, thumb-wheel **1000** and thumb-wheel control logic **1010**.

In its intended use, a message comes to the device via a wireless data communications network, such as the Mobitex™ network, into subsystem **100**, where it is demodulated via DSP **200**, decoded and presented to microprocessor **300** for display on display **500**. To access the display of the message, the user may choose from functions listed under a menu presented as a result of user interaction with thumb-wheel **1000**. If the message is an email message, then the user may choose to respond to the email by selecting "Reply" from a menu presented on the display through interaction via thumb-wheel **1000** or via menu selection from keyboard **900**. In typing the reply, the user can use keyboard **900** to type full text message replies, or insert pre-determined or "canned" responses by using either a particular keystroke pattern or through pulling down pre-determined text strings from a menu of items presented on display **500** through the use of thumb-wheel **1000**.

When the reply to the message is composed, the user can initiate the sending of the message preferably by interaction through thumb-wheel **1000**, or alternatively, with less efficiency, through a combination of keyboard **900** keystrokes. When the microprocessor **300** receives an indication that the message is to be sent, it processes the message for transport, and by directing and communicating with transmitter/receiver subsystem **100**, enables the reply message to be sent via the wireless data communications network to the intended recipient. Similar interaction through I/O devices keyboard **900** and thumb-wheel **1000** can be used to initiate full-text messages or to forward messages to another party.

In addition, the keyboard **900** and thumb-wheel **1000** can be used to permit data entry to an address book resident on the messaging device, or an electronic calendar or log book, or any other function on the messaging device requiring data entry. Preferably, the thumb-wheel is a thumb-wheel with a push button SPST switch with quadrature signal outputs, such as that manufactured by Matsushita Electronic Components Co. Ltd. as part number EVQWK2001, but may, alternatively, be some other type of auxiliary input device.

FIG. **2** is a front view of an exemplary messaging device having a keyboard that is optimized for use with the thumbs. Shown in FIG. **2** are a plurality of letter keys **901**, specialized keys **902**, **903**, **904** and **905**, and a space bar **906**. Also shown is the thumb-wheel **1000** in its vertical orientation and in association with display **500** and keyboard **900**. The specialized key **902** may, for example, be an alt key, the specialized key **903** may be a shift or capitalization key, the specialized key **904** may be a line feed key, and the specialized key **905** may be a backspace key. It should be understood, however, that the specialized keys **902–905** may provide other functions, such as an escape key, a delete key, a home key, a menu key, a cursor-left key or a cursor-right key. It should also be understood, that the messaging device may include additional functional keys. In addition, in alternative embodiments described below, certain functional keys, such as an alt key and shift/cap key, may be positioned on another device surface in addition to or instead of on the face of the device, to be operated, for example, by the fingers or part of the hand of a user.

FIG. **2** also shows the arrangement of keys on the keyboard into multiple rows. Each of the rows are arranged in a concave pattern, such as an arc. In the illustrated embodiment, the rows of keys are arranged along an arc in a concave-down pattern. In other embodiments, however, the rows of keys may be arranged in other concave patterns. For example, the concave pattern may be defined along two intersecting line segments instead of along an arc, and may be a concave-up pattern instead of a concave-down pattern. Such an arrangement of the keys not only facilitates thumb typing by a user but also reduces the space occupied by the keyboard. The concave rows of keys shown in FIG. **2** allow for location of the space bar **906** in its conventional keyboard position but reduce the amount of unoccupied space at the ends of the space bar.

Although FIG. **2** shows a preferred embodiment of a messaging device, other implementations incorporating alternate device architectures are also contemplated. For example, different patterns of the concave rows could be employed to accommodate keys on the keyboard between rows either in addition to or instead of only at the bottom of the keyboard in the position of the space bar shown in FIG. **2**. In addition, many different shapes and orientations of the keys could also be utilized, as is further detailed below with respect to FIGS. **5–10**. Similar keyboard layouts would also be suitable for use in other electronic devices with different display arrangements. Electronic devices having clamshell type designs in which the display is positioned on a movable cover portion of the device which is attached to the keyboard portion with a hinge, represent one such alternate keyboard/display arrangement. It is well within the scope of the present invention to include the inventive keyboard on a variety of handheld electronic devices such handheld electronic arcade devices; two-way pagers; wireless data communication devices; cell phones; and Personal Digital Assistants (PDAs).

In one alternative embodiment, the messaging device may include a light source, such as a backlight, that can be activated by a user of the device to light the keyboard and/or the display in low-light conditions.

FIG. **3** is a view of a subset of the letter keys **901** shown in FIG. **2**, illustrating exemplary dimensions and relative positions of the keys. Also shown is the point **950** that marks the center of keyboard **900**, key dimensions **970**, **971**, **972** and **973**, as well as angle **960** and the rho value **965**, representing curvature of a letter key **901**. In investigating optimal key placement on the keyboard, it was determined that the keys should be placed at an angle **960** relative to a vertical reference bisecting the key that facilitated easy typing using thumbs. That angle is preferably positive 40 degrees relative to the vertical reference for keys on the right side of the keyboard (where **950** is the center of the keyboard), and negative 40 degrees for the keys on the left side of the keyboard. Complementary angles ranging from 20 degrees to 70 degrees could also be used to accomplish the goal, albeit less optimally, of facilitating thumb typing.

It should be understood, however, that alternative key dimensions and/or placements could also be utilized. For instance, the keys on the right and left sides of the keyboard could be tilted at the same angle **960** from vertical (i.e., all of the keys may have a positive angle **960** or all of the keys may have a negative angle **960**), or could all be aligned with the vertical reference (i.e., having no angle **960** from vertical). It should also be understood that the phase "tilted at the same angle" as used within this application means either tilted at equal angles or tilted at nearly equal angles.

As is also shown on FIGS. **2** and **3**, the keys **901** are dispersed across keyboard **900** evenly so that there is sufficient space between the keys to decrease the opportunity for multiple keys being depressed while thumb typing. Additionally, the keys **901** are sized appropriately given the footprint of the messaging device and the keyboard **900**. In one embodiment, the messaging device measures 64 mm by 89 mm across its face, which does not leave much room for keyboard **900** and display **500**. In this embodiment, keyboard **900** occupies over half of the face of the messaging device.

In order to maximize the surface area of the key that a thumb would hit, the keys are preferably oval, and have a rho **965** defining the curvature of the key of 0.414, although values may range higher or lower. Other rho values will lead to an acceptable, but not as optimal, or aesthetically pleasing, shape of keys **901**. As to the key dimensions, the width **970** of the key **901** is 4.8 millimeters (**971** representing the radius of half that value, 2.4 mm) and the length (or height) **972** of the key **901** is 7 millimeters (**973** representing the radius of half that value, 3.5 mm). It should be understood, however, that other key shapes could also be utilized, such as the key shapes illustrated in FIGS. **8**–**10**.

One of the software features that aids in the device being optimally used for thumb typing is a capitalization feature. Using this feature, if a user depresses a key **901**, then the operating system detects a key down event. If the key is released after a period of time, the operating system detects a key up event. If, upon a key down event, a period of time elapses before a key up event is detected, then the operating system determines that a key repeat event has occurred representing a situation where a user has continued to depress a key without releasing it. A key repeat event is then treated by application software residing in either FLASH **600** or RAM **700** as an event that requires the capitalization of the key previously depressed. This feature disables a key repeat feature and substitutes instead a capitalization feature based upon a key repeat. The timing of the key scanning to determine whether a key has been released can be set to permit a slower keyboard response or a faster keyboard response, depending upon user experience or preferences. Depression of a letter key while or immediately after the shift/cap key **903** is depressed may also cause the upper case letter to be entered.

Although the capitalization function preferably works only to change the state of a letter to a capital, it alternatively could operate to change a capital letter to a lower case letter. The actual display is changed by the application program substituting the value of the capital letter in the register that holds the value of the letter to be displayed. As alternatively implemented, the continued depressing without release of a letter key could result in a key oscillating between upper case and lower case, depending on the length of time the key is depressed.

FIG. **4** is the logic circuitry **1010** associated with the thumb-wheel **1000** of FIGS. **1** and **2**. Thumb-wheel **1000** outputs quadrature signals phase A **1021** and phase B **1022**, which are processed by D flip-flops **1031** and **1032** to present signals **1041** W_UP and **1042** W_DN to microprocessor **300**. Signals **1041** and **1042** represent, respectively, a user rolling the thumb-wheel up and rolling the thumb-wheel down.

FIG. **5** is a diagram showing one exemplary embodiment of a keyboard that is optimized for use with the thumbs. This keyboard includes a plurality of letter keys **901** (A–Z), several function keys **902**, **903**, **904**, and a spacebar/symbol selector **906**. The respective keys **901** are preferably organized into three concave rows **910**, **912**, **914**. The first concave row **910** includes the function keys **902**, **903**, and the letter keys **901** Z, X, C, V, B, N, and M, just like on the first row of a standard QWERTY keyboard. The degree of arcing of the first concave row **910** is preferably defined by a first radius **R1**. The second concave row **912** includes the function key **904**, and the letter keys **901** A, S, D, F, G, H, J, K and L, just like on the second row of a standard QWERTY keyboard. The degree of arcing of the second concave row **912** is preferably defined by a second radius **R2**. The third concave row **914** includes the letter keys **901** Q, W, E, R, T, Y, U, I, O and P, just like on the third row of a standard QWERTY keyboard. The degree of arcing of the third concave row **914** is preferably defined by a third radius **R3**.

For the three-row organization shown in FIG. **5**, the first radius R1 of the first concave row **910** is preferably less than the second radius R2 of the second concave row **912**, which is preferably less than the third radius R3 of the third concave row **914**. These radii R1, R2, R3 may define a set of concentric circles on which the concave rows of keys **910**, **912**, **914** are placed. Other configurations and orientations of the concave rows of keys are also possible.

Also shown in FIG. **5** are a set of vertical references **916**. Each of these vertical references **916** bisects one or more (up to three) of the letter keys **901** making up the keyboard. In the embodiment shown in FIG. **5**, the keys **901** are oval-shaped, and are oriented at an angle with respect to the vertical references **916**. The keys on the right-hand side of the keyboard are oriented at a first predetermined angle **918**, and the keys on the left-hand side of the keyboard are oriented at a second predetermined angle **920**. The first predetermined angle **918** is a positive angle with respect to the vertical reference **916**, and the second predetermined angle **920** is a negative angle with respect to the vertical reference **916**. These first and second predetermined angles **918**, **920** are complementary angles, for example +/–40 degrees from vertical. As discussed above, the keys may also be aligned at other angles, all tilted at the same angle, or aligned with the vertical reference.

The keys **901** in FIG. **5** are also preferably aligned along the set of vertical references **916**, such that a key in the first row is aligned with a key in the second row, which is aligned with a key in the third row. For example, the N key in the first row **910** is aligned with the J key in the second row **912** and the U key in the third row **914**. That is, a vertical line **916** drawn through the center of any of the keys **901** in the rows of keys perpendicular to the top and bottom edges of the device will intersect the center of a key in an adjacent row of keys.

FIG. **6** is a diagram showing another exemplary embodiment of a keyboard optimized for use with the thumbs. This embodiment is similar to FIG. **5**, except that the keys **901** in each concave row **910, 912, 914** are not aligned along the set of vertical references **916**, but instead are offset from one row to the next. Keys positioned along a vertical line passing through the center of the keyboard, such as the "G" key in FIG. **6** may be oriented such that an axis of symmetry of the shape coincides with the vertical line passing through the center of the keyboard, thereby allowing the key to be used as easily with the left as the right thumb. In the figure, although the "G" key was oriented with the major axis coinciding with a vertical, it could have been placed with the minor axis coinciding with the vertical. In another embodiment as shown in FIG. **7**, a special center key **922** has a vertically symmetrical key shape and orientation that is a combination of the left key shape and the right key shape: by superimposing the two shapes and tracing the exterior circumference as a central shape, the resulting shape can be used just as easily with the left or right thumb.

FIG. **8** shows a keyboard embodiment with circular keys. This embodiment is similar to FIGS. **6** and **7**, except that the keys **901** in each concave row **910, 912, 914** are circular in shape instead of ovals. Because of the circular shape of the keys **901**, the concept of orienting the keys **901** at the first and second predetermined angles **918, 920** is not applicable to this design. However, the concept of a circumscribed oval still applies, as in the case of a central key discussed above in reference to FIG. **6**. It is possible to circumscribe a thumb-impact oval onto the keys with a major axis coinciding with a line going through the center of each circular key at an angle **918** and **920** for right sided keys and left sided keys respectively. Note that the keys **901** in FIG. **8** are arranged in an offset key arrangement. In an alternative embodiment, the circular keys could be also aligned along a set of vertical references, similar to FIG. **5**.

FIG. **9** shows a keyboard embodiment with square keys. This embodiment is similar to the embodiments shown in FIGS. **5** and **8**, except that the keys are square instead of ovals or circles. The keys in FIG. **9** are aligned along the set of vertical references **916**. In alternative embodiments, however, the square keys may instead be tilted, offset, or both.

FIG. **10** shows a keyboard embodiment having rectangular keys. This embodiment is similar to the embodiment shown in FIG. **9**, except that the keys **901** in each concave row **910, 912, 914** are rectangular, and are slanted similar to the keys described above with reference to FIG. **5**. In one alternative embodiment, the rectangular keys may be arranged in an offset layout, with a center key having a vertically symmetrical key shape and orientation that is a combination of the left key shape and the right key shape, such that the resulting shape can be used just as easily with the left or right thumb.

FIGS. **11***a* through **11***c* show front and side views of a hand-held electronic device incorporating an alternative functional key arrangement. As described above, a keyboard optimized for use with the thumbs may comprise keys which will normally be operated with either the right or the left thumb of a user, as well as possibly one or more keys that may be operated with either thumb. In the case of certain

functional keys such as an alt key **902** or a shift key designated **902***a* in FIG. **11***a*, a device input is made when a letter key is depressed simultaneously with or subsequent to the operation of the functional key. For example, the alt key **902** may be operated in order to input a number or symbol associated with a letter key **901**. In the example keyboard of FIG. **5**, operation of the 'Q' key will normally cause a lowercase 'q' to be input to the device. The number '1' may be input when the 'Q' key is operated while or after the alt key **902** is operated. Similarly, an uppercase 'Q' could be entered when the 'Q' key is depressed while the shift key **902***a* is depressed or immediately after the shift key has been depressed.

Since the space that the keyboard occupies is to be minimized however, only a single alt key **902** and a single shift key **902***a* can be accommodated on a small device. Thus, a user's thumb typing may be interrupted when a letter key that is normally operated using the same thumb used to operate the alt key **902** or the shift key **902***a* is to be operated in conjunction with the alt key or shift key. According to an aspect of the invention, the device **1100** in FIG. **11***a* includes the functional keys **1102** and **1104**, which are located on different surfaces of the device housing (on the sides) than the keyboard and are thus operable using other parts of the hand such as a finger or an inside part of the hand. The functional keys **1102** and **1104** may therefore easily be operated in conjunction with the depression of letter keys by the thumbs without interrupting a user's typing.

Where the key **1102** is the alt key for example, to enter the number '1' a user need simply press the functional key **1102** using a finger or part of the hand instead of a thumb and then depress the 'Q' key (see FIG. **5** for example). If the key **1104** is a shift key, then when an upper case letter is to be entered, the user may depress the key **1104** using a finger or part of the hand. Thus, the thumbs may be used only for operation of letter keys **901** and keys requiring operation of a letter key in conjunction therewith are operated using other parts of the hand. Typing speed may thereby be improved in that the entry of an upper case letter, a number, a symbol or any other special character associated with a letter key **901** does not require operation of a functional key with the thumbs.

It should be understood that, although two functional keys **1102** and **1104** are shown in FIGS. **11***a* to **11** *c*, fewer or more than two functional keys may be positioned on the device for operation with other parts of a user's hands. The number of functional keys may depend, for example, upon the particular keyboard key assignments and the relative expected frequency of use of a functional key. Where device software automatically capitalizes the first word of a sentence for example, a shift key might not be used particularly often, such that only an alt key might be provided at position **1102** or **1104**. In this example and if desired, both keys **1102** and **1104** may be alt keys, such that a user may use either hand to invoke the alt function.

When a functional key is positioned for operation using another part of the hand than the thumb, the functional key need not necessarily also be provided on the keyboard, thereby reducing the space occupied by the keyboard. However, in order to provide a more familiar interface for a user, the keyboard functional keys such as **902** and **902***a* may be maintained. A user then has the option to use either the thumb-operated keyboard functional keys **902, 902***a* or the finger-or-hand-operated functional keys **1102, 1104**. Alternatively, if keys **1102** and **1104** are for example alt and shift keys, then the keyboard alt and shift keys **902** and **902***a* may be assigned other functions or inputs to thereby further expand keyboard functionality.

FIGS. **12***a* through **12***c* are diagrams showing front and side views of a hand-held electronic device incorporating another alternative functional key arrangement. The device **1200** is substantially similar to the device **1100** except that functional keys **1202** and **1204** are positioned on the sides of the device housing toward the bottom of the device for operation by a part of the hand instead of the thumbs or fingers. As described above, the functional keys **1202** and **1204** may be operated by a user using a part of the hand, such as the palm of the hand or part of the band near the base of the fingers depending upon how tbe user holds the device when in use, while typing on the keyboard with the thumbs. As also described above, more or fewer than the two functional keys **1202** and **1204** may be provided on the device, and the keys **1202** and **1204** may implement different functions or the same function, and may be provided instead of or in addition to the keyboard functional keys **902** and **902***a*.

FIGS. **13***a* and **13***b* show front and top views of a hand-held electronic device incorporating a functional key arrangement according to another embodiment of the invention. In this embodiment, functional keys **1302** and **1304** are provided at the top of the housing of device **1300**, for operation using the fingertips. As above, more or fewer than the two functional keys **1302**, **1304** shown in FIGS. **13***a* and **13***b* may be provided in addition to or instead of keyboard functional keys **902**, **902***a*, and such keys may be used to invoke either the same function or different functions.

FIGS. **14***a* and **14***b* sbow rear views of a hand-held electronic device in which respective functional key arrangements are implemented. In FIG. **14***a*, two functional keys are shown: an alt key **1402** and a shift key **1404**. These particular functional keys are labeled for illustrative purposes only and as such are not intended to limit the scope of the invention. Other functions may also or instead be associated with the functional keys **1402** and **1404**. The keys **1402** and **1404** are preferably located so as not to interfere with any removable device elements such as a removable housing section **1406** which may, for example, cover a battery compartment. Rear-mounted keys are also preferably positioned to provide ample space for a user to hold the device without contacting the functional keys **1402**, **1404**.

In the FIG. **14***a* arrangement, the alt key **1402** is positioned for operation with a finger of the left hand of the user, whereas the shift key **1404** is positioned for operation with a finger of the right hand. Where a user is familiar with a keyboard layout as shown in FIG. **2** for example, although the keys **1402**, **1404** are hidden from view when a device **1400** is in use, the associations between a left band alt operation and a right hand shift operation are maintained. However, the invention is in no way restricted to this specific key designation and relative positioning.

One alternative functional key arrangement is shown in FIG. **14***b*, wherein a shift key and an alt key are provided for operation with the fingers of each hand. The shift and alt key pair **1412**, **1414** would be operable using one or more fingers of the left band, and the pair **1416**, **1418** are operable using fingers of the right hand. Although four separate keys are shown in FIG. **14***b*, two centrally positioned and possibly elongated keys accessible to the fingers of both hands when a device **1410** is in use could replace the four key arrangement. With either of these arrangements, the relative shift/alt key positioning, i.e., a shift key above the alt key and both keys operable using either hand, will be familiar to users of a PC keyboard.

It should be understood that the above functional key embodiments are not mutually exclusive. A hand-held electronic device may possibly be provided with functional keys on its sides, top, back or any combination thereof. A user may then have the option of using whichever function key set he or she finds easiest to use. It is also contemplated that devices may incorporate different functional key arrangements or combinations thereof, allowing a user to choose a particular functional key arrangement when a device is purchased.

FIGS. **15***a* to **15***c* show front and side views of an alternate hand-held electronic device incorporating embodiments of the invention. The device **1500** may be similar to the devices **1100**, **1200**, **1300**, **1400** and **1410** except that its display screen **1501** is substantially larger and its keyboard is somewhat different. Provided that a device keyboard includes at least one functional key **902** or functional keys are desired in order to expand functionality of a device keyboard without adding keyboard keys, then the present invention may be particularly advantageous. In FIGS. **15***a* to **15***c*, the functional keys may be positioned, for example, as shown at **1502**, **1504**, **1506**, **1508**, **1510**, **1512**, or possibly at more than one such location, for operation using the fingers or part of the hand substantially as described above.

FIG. **16** is a rear view of a device, such as shown in FIG. **15**, in which a functional key arrangement according to a further embodiment of the invention is implemented. As shown in FIG. **16**, functional keys such as shift keys **1602**, **1606** and alt keys **1604** and **1608** may be positioned at locations on a rear device housing so as to be accessible to a user's fingers while providing sufficient space for a user to hold the device without contacting the functional keys and not interfering with any removable device housing sections or other internal or external device components.

Depending upon the size of the device **1500**, a user might not be able to use top-mounted keys such as shown in FIGS. **13***a* and **13***b* while thumb typing on the keyboard. Side-and back-mounted functional keys as shown in FIGS. **15***a* to **15***c* and **16** are therefore more suited to devices having larger form factors than top-mounted keys. It should be understood, however, that whenever device physical dimensions permit, top-mounted functional keys may be implemented.

Although the devices shown in FIGS. **11***a–c*, **12***a–c*, **13***a–b* and **15***a–c* include slanted oval-shaped keyboard keys, the functional keys and arrangements disclosed herein are in no way restricted to this type of keyboard key shape or layout. The invention may be implemented in conjunction with any of the keyboards described above, including those shown in FIGS. **2** and **5**–**10** for example, as well as virtually any other hand-held electronic device keyboard.

Similarly, the present invention is not restricted to devices having a "full" keyboard. FIGS. **17***a* to **17***d* show front, side and top views of a further alternative hand-held electronic device incorporating functional key arrangements according to embodiments of the invention. The device shown in FIGS. **17***a*–**17***d* is a mobile telephone **1700**, having a standard numeric keypad **1702** and a plurality of additional keys **1704**, **1706**, **1708** and **1710**, which may include, for example, a power key, a send key, an end key, and like keys typically found on mobile telephones. Mobile telephone users normally operate such keys using the thumb of one hand while holding the telephone. The device **1700** also includes a microphone **1712**, a speaker **1714** and a display screen **1715**. In accordance with the invention, one or more of the functional keys **1716**, **1718**, **1820**, **1722**, **1724**, **1726** and **1728** are implemented in the device **1700**. It should be understood that mobile telephones are available in many different forms and sizes, and that the present invention is applicable to virtually any design. For example, a device **1700** may have a "slim" form factor, wherein its side profile or depth dimension is significantly smaller than its front/back profile or width, or comprise a clamshell design with hinged sections. Any functional keys **1716–1728** could then be sized and positioned accordingly.

When used only to enter numbers, a keypad **1702** is normally sufficient. However, text entry via the keypad is often required, and is becoming much more common with the increasing popularity of Short Message Service (SMS) and other text-or data-related functions that mobile telephones may support. Although numeric keypad keys also have associated text characters, known text entry schemes involving multiple depressions of a single key for example tend to be slow, even when used in conjunction with predictive automatic text techniques. Therefore, one contemplated function for a functional key is to select between numeric and text entry when a keypad key is depressed. For example, if a user wishes to enter the letter 'A', normally associated with the number key '2' on standard telephone keypads, the user could depress a function key, **1716** for example, and simultaneously depress the '2' key. The functional key **1716** may be held in its depressed position as long as text entry is required. When the key **1716** is released, the keypad then preferably reverts back to a numeric entry state. Alternatively, a single depression of the functional key **1716** may toggle the keypad between numeric and text entry states. Such functionality may provide for much faster and easier text entry on a substantially thumb-operated numeric keypad.

Where more than one functional key is provided on the device **1700**, text entry may be further facilitated by allowing a user to select between the multiple characters associated with a keypad key. In a device **1700** with four keys for example, any particular letter associated with any keypad key might be selected using the function keys. Operation of a first functional key in conjunction with a '7' key on a conventional keypad might select a first text character 'P', whereas second, third and fourth functional keys may be used to respectively select 'Q', 'R' and 'S'. As described above, any functional key and keypad key could preferably be operated simultaneously, using a finger or part of the hand to operate a functional key while using a thumb to depress a keypad key.

It should be understood that the above text entry function is merely an illustrative example of a possible implementation of an embodiment of the invention. Other functions allowing expansion of keypad key functionality through the use of finger-or-hand-operated functional keys are also within the scope of the present invention.

As described above for the preceding keypad arrangements, the functional keys **1716–1728** may provide the same or different functions. In devices such as device **1700** which are normally held on one hand, duplication of functional keys on each side of the device may be particularly advantageous in that the device may be used with either a left hand or a right hand. The left-hand side functional keys **1716–1720** would be operable using fingers on the right hand, left-hand side functional keys **1722–1726** would be operable using fingers on the left hand, and the top functional key **1728** would be accessible by either hand.

In one alternative embodiment the functional keys shown in FIGS. **17*a–d*** may also or instead be provided on the rear of device **1700**.

This written description uses examples to disclose the invention, including the best mode, and also to enable any person skilled in the art to make and use the invention. The patentable scope of the invention is defined by the claims, and may include other examples that occur to those skilled in the art.

For example, the functional keys have been shown in the drawings as rectangular keys. Implementation of functional keys having other shapes intended to improve key operability or esthetic appeal of a device are also contemplated. Functional keys on the same device might also have different shapes adapted to the part of the fingers or hands by which the keys will be operated or to allow a user to distinguish between different functional keys. Similarly, although the functional keys have been shown in the drawings as projecting beyond the device housing, the keys may instead be substantially flush with or recessed below the device housing surface in order to prevent inadvertent operation thereof.

In addition, the alt and shift functional keys are shown in some of the drawings for illustrative purposes only. Other functional keys that are normally operated in conjunction with other letter keys, such as a control (ctrl) keyboard key, may also or instead be implemented in accordance with the invention.

We claim:

1. A hand-held messaging device, comprising:

a miniaturized QWERTY style keyboard that is horizontally positioned symmetrically between a left edge and a right edge of a face of the hand-held messaging device and having a plurality of keys arranged in three rows across the face, wherein each row of keys is arranged in a concave pattern and wherein each of the plurality of keys is aligned along a vertical axis with a key from an adjacent row of keys, the vertical axis extending perpendicularly from a top edge to a bottom edge of the hand-held messaging device;

a display that is vertically positioned between the keyboard and a top edge of the face and horizontally positioned symmetrically between the left edge and the right edge of the face; and

a processor coupled to the keyboard and the display that controls the operation of the hand-held messaging device;

wherein keys that are aligned along a vertical axis are aligned with substantially no vertical offset.

2. The hand-held messaging device of claim **1**, wherein the hand-held messaging device includes a plurality of connecting surfaces for connecting the face to a bottom surface, and further comprising:

at least one functional key positioned on the connecting surfaces of the hand-held messaging device.

3. The hand-held messaging device of claim **2**, wherein the functional key is an alt key.

4. The hand-held messaging device of claim **2**, wherein the functional key is a capitalization key.

5. The hand-held messaging device of claim **1**, wherein the hand-held messaging device includes a plurality of connecting surfaces for connecting the face to a bottom surface, and further comprising:

at least one functional key positioned on the bottom surface of the hand-held messaging device.

6. The hand-held messaging device of claim **5**, wherein the functional key is an alt key.

7. The hand-held messaging device of claim **5**, wherein the functional key is a shift key.

8. The hand-held messaging device of claim **1**, wherein the hand-held messaging device includes a plurality of connecting surfaces for connecting the face to a bottom surface, and further comprising:

at least one type of functional key, wherein two of each type of functional key are positioned on the bottom surface of the hand-held messaging device.

9. The hand-held messaging device of claim 8, wherein the two functional keys of each type include a first functional key that is positioned for use with a left hand of a user and a second functional key that is positioned for use with a right hand of the user.

10. The hand-held messaging device of claim 1, wherein the hand-held messaging device includes a plurality of connecting surfaces for connecting the face to a bottom surface, and further comprising:

at least one type of functional key, wherein two of each type of functional key are positioned on two of the connecting surfaces, respectively.

11. The hand-held messaging device of claim 10, wherein the two functional keys of each type include a first functional key that is positioned for use with a left hand of a user and a second functional key that is positioned for use with a right hand of a user.

12. The hand-held messaging device of claim 1, wherein the hand-held messaging device includes a plurality of connecting surfaces for connecting the face to a bottom surface, and further comprising:

at least one type of functional key, wherein two of each type of functional key are positioned on one of the connecting surfaces.

13. The hand-held messaging device of claim 12, wherein the two functional keys of each type include a first functional key that is positioned for use with a left hand of a user and a second functional key that is positioned for use with a right hand of user.

14. The hand-held messaging device of claim 1, wherein each row of keys is arranged in a concave-up pattern.

15. The hand-held messaging device of claim 1, wherein each row of keys is arranged in a concave-down pattern.

16. The hand-held messaging device of claim 1, wherein each row of keys is arranged along an arc.

17. The hand-held messaging device of claim 1, wherein each row of keys is arranged along two intersecting line segments.

18. The hand-held messaging device of claim 1, wherein at least one of the plurality of keys of the keyboard is oblong.

19. The hand-held messaging device of claim 18, wherein the oblong key is tilted at an angle from a vertical axis extending through a center of the key.

20. The hand-held messaging device of claim 18, wherein the oblong key is aligned with a vertical axis extending through a center of the key.

21. The hand-held messaging device of claim 1, wherein the plurality of keys of the keyboard are oblong.

22. The hand-held messaging device of claim 21, wherein a first portion of the oblong keys are tilted at a negative angle from vertical and a second portion of the oblong keys are tilted at a positive angle from vertical.

23. The hand-held messaging device of claim 21, wherein the oblong keys are all tilted at the same or substantially the same angle from vertical.

24. The hand-held messaging device of claim 1, wherein at least one of the plurality of keys of the keyboard is oval.

25. The hand-held messaging device of claim 24, wherein the oval key is tilted at an angle from a vertical axis extending through a center of the key.

26. The hand-held messaging device of claim 24, wherein the oval key is aligned with a vertical axis extending through a center of the key.

27. The hand-held messaging device of claim 1, wherein at least one of the plurality of keys of the keyboard is circular.

28. The hand-held messaging device of claim 1, wherein at least one of the plurality of keys of the keyboard is rectangular.

29. The hand-held messaging device of claim 28, wherein the rectangular key is tilted at an angle from a vertical axis extending through a center of the key.

30. The hand-held messaging device of claim 28, wherein the rectangular key is aligned with a vertical axis extending through a center of the key.

31. The hand-held messaging device of claim 1, wherein the plurality of keys each have a shape that is contoured for optimal typing with a user's thumbs.

32. The hand-held messaging device of claim 1, wherein the keyboard includes:

a first row of keys having ten (10) keys, wherein a first set of five (5) keys from the first row are arranged in a pattern having a positive slope from vertical and a second set of five (5) keys from the first row are arranged in a pattern having a negative slope from vertical;

a second row of keys having nine (9) keys, wherein a first set of four (4) keys from the second row are arranged in a pattern having a positive slope from vertical, a second set of four (4) keys from the second row are arranged in a pattern having a negative slope from vertical, and a center key from the second row is positioned equidistant or substantially equidistant from the left edge and the right edge of the face; and

a third row of keys having at least eight keys, wherein a first set of four (4) keys from the third row is arranged in a pattern having a positive slope from vertical and a second set of four (4) keys from the third row is arranged in a pattern having a negative slope from vertical.

33. The hand-held messaging device of claim 32, wherein one of the keys from the third row is a line feed key.

34. The hand-held messaging device of claim 1, wherein the keyboard includes three (3) rows of keys, wherein each of the three rows of keys includes a first set of five (5) keys that are arranged in a pattern having a positive slope from vertical and a second set of five (5) keys that are arranged in a pattern having a negative slope from vertical.

35. The hand-held messaging device of claim 1, wherein the keyboard includes twenty-six (26) letter keys.

36. The hand-held messaging device of claim 35, wherein the twenty-six (26) letter keys are arranged in the format of a QWERTY-style keyboard.

37. The hand-held messaging device of claim 36, further comprising:

a row of functional keys that are horizontally positioned symmetrically or substantially symmetrically between a left edge and a right edge of the face of the hand-held messaging device and vertically positioned between the keyboard and a bottom edge of the hand-held messaging device.

38. The hand-held messaging device of claim 37, wherein the row of functional keys includes a space bar.

39. The hand-held messaging device of claim 37, wherein the row of functional keys includes a an alt key, and wherein at least one letter key has an associated alternate character that may be input to the processor by simultaneously depressing the letter key and the alt key.

40. The hand-held messaging device of claim 37, wherein the row of functional keys includes a shift key.

41. The hand-held messaging device of claim **37**, wherein the row of functional keys includes a menu key.

42. The hand-held messaging device of claim **37**, further comprising:

at least one additional functional key positioned below the row of functional keys.

43. The hand-held messaging device of claim **42**, wherein the additional functional key is a scroll-up key.

44. The hand-held messaging device of claim **42**, wherein the additional functional key is a scroll-down key.

45. The hand-held messaging device of claim **42**, wherein the additional functional key launches an application program.

46. The hand-held messaging device of claim **1**, further comprising:

at least one additional functional key positioned above the keyboard.

47. The hand-held messaging device of claim **46**, wherein the additional functional key is a backspace key.

48. The hand-held messaging device of claim **46**, wherein the additional functional key is a home key.

49. The hand-held messaging device of claim **46**, wherein the additional functional key is an escape key.

50. The hand-held messaging device of claim **46**, wherein the additional functional key is a menu key.

51. The hand-held messaging device of claim **46**, wherein the additional functional key is a delete key.

52. The hand-held messaging device of claim **46**, wherein the additional functional key is a cursor-left key.

53. The hand-held messaging device of claim **46**, wherein the additional functional key is a cursor-right key.

54. The hand-held messaging device of claim **1**, further comprising:

a thumb-wheel coupled to the processor, wherein the thumb-wheel is vertically positioned on the face of the hand-held messaging device above the keyboard and below the display.

55. The hand-held messaging device of claim **1**, wherein the thumb-wheel is horizontally positioned symmetrically or substantially symmetrically between the left edge and the right edge of the face such that the thumb-wheel may be operated by a user with either a right hand or a left hand.

56. The hand-held messaging device of claim **1**, further comprising:

a thumb-wheel coupled to the processor, wherein the thumb-wheel is positioned on a side surface of the hand-held messaging device.

57. The hand-held messaging device of claim **1**, further comprising:

a rocker switch coupled to the processor, wherein the rocker switch is positioned on a side surface of the hand-held messaging device.

58. The hand-held messaging device of claim **1**, further comprising:

a wireless radio subsystem coupled to the processor that transmits and receives electronic messages from a wireless network; and

a memory device coupled to the processor that stores electronic messages received from the wireless network.

59. The hand-held messaging device of claim **58**, further comprising:

application software executing on the processor, wherein the application software includes an electronic messaging application that receives electronic messages that are wirelessly redirected to the hand-held messaging device from a redirection software application executing on a corporate server.

60. The hand-held messaging device of claim **59**, wherein the application software includes a calendar application.

61. The hand-held messaging device of claim **1**, further comprising:

a rechargeable battery coupled to the processor that supplies power to the hand-held messaging device.

62. The hand-held messaging device of claim **1**, further comprising:

a light source positioned to provide light to the keyboard.

63. The hand-held messaging device of claim **62**, wherein the light source also provides light to the display.

64. The hand-held messaging device of claim **62**, wherein the light source is a backlight mounted within a housing of the hand-held messaging device.

65. A hand-held messaging device, comprising:

a device housing having a face, a bottom surface, and a plurality of connecting surfaces for connecting the face to the bottom surface;

a display mounted within the face of the device housing and horizontally positioned symmetrically between a left edge of face and a right edge of the face;

a miniaturized keyboard mounted within the face of the device housing in a position between the display and a bottom edge of the face, wherein the keyboard comprises a QWERTY-stile keyboard having a plurality of keys arranged in three rows across the face, wherein each row of keys is arranged in a concave pattern and is distributed symmetrically across the face of the housing, wherein the keyboard includes a plurality of letter keys and at least one specialized key and wherein each of the plurality of keys is aligned along a vertical axis with a key from an adjacent row of keys, the vertical axis extending perpendicularly from a top edge to the bottom edge of the face of the device housing; and

an auxiliary input device mounted within the housing;

wherein keys that are aligned along a vertical axis are aligned with substantially no vertical offset.

66. The hand-held messaging device of claim **65**, further comprising:

at least one functional key mounted within the connecting surfaces of the device.

67. The hand-held messaging device of claim **65**, further comprising:

at least one functional key mounted within the bottom surface of the device.

68. The hand-held messaging device of claim **65**, wherein each row of keys is arranged in a concave-up pattern.

69. The hand-held messaging device of claim **65**, wherein each row of keys is arranged in a concave-down pattern.

70. The hand-held messaging device of claim **65**, wherein each row of keys is arranged along an arc.

71. The hand-held messaging device of claim **65**, wherein each row of keys is arranged along two intersecting line segments.

72. The hand-held messaging device of claim **65**, wherein the plurality of keys are oblong.

73. The hand-held messaging device of claim **72**, wherein the oblong shaped keys are tilted with respect to a vertical reference through the face of the device housing.

74. The hand-held messaging device of claim **72**, wherein the oblong shaped keys are aligned with a vertical reference through the face of the device housing.

75. The hand-held messaging device of claim **72**, wherein the oblong shaped keys are oval shaped.

76. The hand-held messaging device of claim **65**, wherein the plurality of keys are circular.

77. The hand-held messaging device of claim **65**, wherein the plurality of keys are square.

78. The hand-held messaging device of claim **65**, wherein the plurality of keys are rectangular.

79. The hand-held messaging device of claim **65**, wherein the specialized key is a line feed key.

80. The hand-held messaging device of claim **65**, wherein the specialized key is a backspace key.

81. The hand-held messaging device of claim **65**, wherein the keyboard further comprises a row of functional keys.

82. The hand-held messaging device of claim **65**, wherein the row of functional keys includes at least a space bar key, an alt key, and a shift key.

83. The hand-held messaging device of claim **81**, wherein the row of functional keys includes at least a space bar key, a shift key, and a menu key.

84. The hand-held messaging device of claim **65**, wherein the hand-held device is a two-way pager, a personal digital assistant or an electronic organizer.

85. The hand-held messaging device of claim **65**, wherein the display occupies more than half of the surface area of the face of the device.

86. The hand-held messaging device of claim **65**, wherein the auxiliary input device is mounted within one of the connecting surfaces.

87. The hand-held messaging device of claim **65**, wherein the auxiliary input device is a thumbwheel.

88. The hand-held messaging device of claim **65**, wherein the auxiliary input device is a rocker switch.

89. The hand-held messaging device of claim **65**, wherein the auxiliary input device includes a directional input component for navigating a plurality of menu items presented on the display and a selector switch for selecting a menu item from the plurality of menu items.

90. The hand-held messaging device of claim **65**, wherein the auxiliary input device is mounted within the face of the device.

91. The hand-held messaging device of claim **65**, further comprising:

a transceiver for transmitting and receiving messages.

92. The hand-held messaging device of claim **91**, further comprising:

a first antenna for receiving messages; and

a second antenna for transmitting messages.

93. The hand-held messaging device of claim **92**, wherein the transceiver further comprises:

a receiver, coupled to the first antenna, for demodulating the received messages; and

a transmitter, coupled to the second antenna, for generating a modulated message.

94. The hand-held messaging device of claim **93**, wherein the transceiver further comprises:

a digital signal processor coupled to the transmitter and the receiver for processing demodulated messages from the receiver, and for providing modulation information to the transmitter.

95. The hand-held messaging device of claim **65**, wherein the keyboard includes a backlight.

96. A wireless e-mail device, comprising:

a device housing having a face and a left and right side surface coupled to the face; display mourned within the face;

a transceiver for receiving e-mail messages from a wireless network and for transmitting e-mail messages generated on the wireless e-mail device to the wireless network; and

a miniaturized QWERTY style keyboard that is horizontally positioned symmetrically between the left side surface and the right side surface and having a plurality of keys arranged in three rows across the face, wherein each row of keys is arranged in a concave pattern and wherein each of the plurality of keys is aligned along a vertical axis with a key from an adjacent row of keys, the vertical axis extending perpendicularly from a top edge to a bottom edge of the face of the device housing;

wherein keys that are aligned along a vertical axis are aligned with substantially no vertical offset.

97. The wireless e-mail device of claim **96**, further comprising:

at least one functional key positioned on the left side surface of the device.

98. The wireless e-mail device of claim **96**, further comprising:

at least one functional key positioned on the right side surface of the device.

99. The wireless e-mail device of claim **96**, further comprising:

at least one type of functional key, wherein the one type of functional key is positioned on both the left and right side surfaces of the device.

100. The wireless e-mail device of claim **96**, wherein the device housing includes a bottom surface coupled to the left and right side surfaces, and further comprising:

at least one functional key positioned on the bottom surface of the housing.

101. The wireless e-mail device of claim **96**, wherein the device housing includes a top side surface coupled to the face, and further comprising:

at least one functional key positioned on the top side surface.

102. The wireless e-mail device of claim **96**, further comprising an auxiliary input device mounted within the device housing.

103. The wireless e-mail device of claim **102**, wherein the auxiliary input device is a thumbwheel.

104. The wireless e-mail device of claim **102**, wherein the auxiliary input device is a rocker switch.

105. The wireless e-mail device of claim **102**, wherein the auxiliary input device includes a directional input component for navigating a plurality of menu items presented on the display and a selector switch for selecting a menu item from the plurality of menu items.

106. The wireless e-mail device of claim **96**, further comprising an antenna coupled to the transceiver.

107. The wireless e-mail device of claim **96**, further comprising:

a microprocessor; and

a memory for storing an operating system and a plurality of application programs that are executed by the microprocessor to control the operation of the wireless e-mail device.

108. The wireless e-mail device of claim **107**, further comprising a digital signal processor coupled between the microprocessor and the transceiver.

109. The wireless e-mail device of claim **107**, wherein the plurality of application programs include a messaging application for generating e-mail messages and a calendar application.

110. The wireless e-mail device of claim **109**, wherein the plurality of application programs further include an address book application.

111. The wireless e-mail device of claim **107**, wherein the memory is a flash memory.

112. The wireless e-mail device of claim **96**, further comprising a serial port for coupling the wireless e-mail device to a host computer.

113. The wireless e-mail device of claim **96**, further comprising a power supply system including a rechargeable battery and an external charger input for receiving a source of electrical charge to recharge the rechargeable battery.

114. The wireless e-mail device of claim **113**, wherein the rechargeable battery is a lithium battery.

115. The wireless e-mail device of claim **113**, wherein the power supply subsystem further includes a voltage regulator coupled to the rechargeable battery for generating a regulated supply voltage for powering the device.

116. The wireless e-mail device of claim **113**, wherein the power supply subsystem further includes connections to a microprocessor for monitoring the operation of the power supply subsystem.

117. A hand-held messaging device, comprising:

a device housing having a face;

a display mounted within the face;

a miniaturized keyboard mounted within the face of the device housing in a position between the display and a bottom edge of the face, wherein the keyboard comprises a QWERTY-style keyboard having a plurality of keys arranged in three rows across the face, wherein each row of keys is arranged in a concave pattern and is distributed symmetrically across the face of the housing, wherein the keyboard includes a plurality of letter keys and at least one specialized key and wherein each of the plurality of keys is aligned along a vertical axis with a key front an adjacent row of keys, the vertical axis extending perpendicularly from a top edge to the bottom edge of the face; and

means for receiving e-mail messages from a wireless network and for transmitting e-mail messages generated on the hand-held messaging device to the wireless network;

wherein keys that are aligned along a vertical axis are aligned with substantially no vertical offset.

118. The hand-held messaging device of claim **117**, wherein the device housing includes a left side surface and a right side surface, and further comprising:

at least one functional key positioned on the left side surface of the device.

119. The hand-held messaging device of claim **117**, wherein the device housing includes a left side surface and a right side surface, and further comprising:

at least one functional key positioned on the right side surface of the device.

120. The hand-held messaging device of claim **117**, wherein the device housing includes a left side surface and a right side surface, and further comprising:

at least one type of functional key, wherein the one type of functional key is positioned on both the left and right side surfaces of the device.

121. The hand-held messaging device of claim **117**, wherein the device housing includes a bottom surface coupled to the left and right side surfaces, and further comprising:

at least one functional key positioned on the bottom surface of the housing.

122. The hand-held messaging device of claim **117**, wherein the device housing includes a top side surface coupled to the face, and further comprising:

at least one functional key positioned on the top side surface.

123. A hand-held messaging device, comprising:

a device housing having a face;

a display mounted within the face;

a miniaturized keyboard mourned within the face of the device housing in a position between the display and a bottom edge of the face, wherein the keyboard comprises a QWERTY-style keyboard having a plurality of keys arranged in three rows across the face, wherein each row of keys is arranged in a concave pattern and is distributed symmetrically across the face of the housing, wherein the keyboard includes a plurality of letter keys and at least one specialized key and wherein each of the plurality of keys is aligned along a vertical axis with a key from an adjacent row of keys, the vertical axis extending perpendicularly from a top edge to the bottom edge of the face;

an auxiliary input device mounted within the device housing; a transceiver for receiving information from a wireless network and for transmitting information to the wireless network;

an antenna coupled to the transceiver

a microprocessor;

a memory for storing an operating system and a plurality of application programs that are executed by the microprocessor to control the operation of the hand-held messaging device; and a power supply system including a rechargeable battery and an external charger input for receiving a source of electrical charge to recharge die rechargeable battery;

wherein keys that arc aligned along a vertical axis are aligned with substantially no vertical offset.

124. The hand-held messaging device of claim **123**, wherein the device housing includes a left side surface and a right side surface, and further comprising:

at least one functional key positioned on the left side surface of the device.

125. The hand-held messaging device of claim **123**, wherein the device housing includes a left side surface and a right side surface, and further comprising:

at least one functional key positioned on the right side surface of the device.

126. The hand-held messaging device of claim **123**, wherein the device housing includes a left side surface and a right side surface, and further comprising:

at least one type of functional key, wherein the one type of functional key is positioned on both the left and right side surfaces of the device.

127. The hand-held messaging device of claim **123**, wherein the device housing includes a bottom surface coupled to the left and right side surfaces, and further comprising:

at least one functional key positioned on the bottom surface of the housing.

128. The hand-held messaging device of claim **123**, wherein the device housing includes a top side surface coupled to the face, and further comprising:

at least one functional key positioned on the top side surface.

* * * * *

JS 44 (Rev. 3/99)

CIVIL COVER SHEET

The JS 44 civil cover sheet and the information contained herein neither replace nor supplement the filing and service of pleadings or other papers as required by law, except as provided by local rules of court. This form, approved by the Judicial Conference of the United States in September 1974, is required for the use of the Clerk of Court for the purpose of initiating the civil docket sheet. (SEE INSTRUCTIONS ON THE REVERSE OF THE FORM.)

I. (a) PLAINTIFFS

MOTOROLA, INC.

(b) County Of Residence Of First Listed Plaintiff:
(Except In U.S. Plaintiff Cases)

(c) Attorneys (Firm Name, Address, And Telephone Number)
Josy W. Ingersoll (No. 1088)
Elena C. Norman (No. 4780)
Monté T. Squire (No. 4764)
Young Conaway Stargatt & Taylor, LLP, The Brandywine Building,
1000 West Street, 17th Floor, Wilmington, DE 19801

DEFENDANT

RESEARCH IN MOTION LIMITED AND RESEARCH IN MOTION CORPORATION,

County Of Residence Of First Listed Defendant:
(IN U.S. PLAINTIFF CASES ONLY)
NOTE: IN LAND CONDEMNATION CASES, USE THE LOCATION OF THE TRACT OF LAND INVOLVED

Attorneys (If Known)

II. BASIS OF JURISDICTION (PLACE AN X IN ONE BOX ONLY)

□ 1 U.S. Government Plaintiff

□ 2 U.S. Government Defendant

☒ 3 Federal Question (U.S. Government Not a Party)

□ 4 Diversity (Indicate Citizenship of Parties in Item III)

III. CITIZENSHIP OF PRINCIPAL PARTIES (Place An X In One Box For Plaintiff And One Box For Defendant)
(For Diversity Cases Only)

	PTF	DEF		PTF	DEF
Citizen of This State	□ 1	□ 1	Incorporated or Principal Place of Business in This State	□ 4	□ 4
Citizen of Another State	□ 2	□ 2	Incorporated and Principal Place of Business in This State	□ 5	□ 5
Citizen or Subject of a Foreign Country	□ 3	□ 3	Foreign Nation	□ 6	□ 6

V. NATURE OF SUIT (Place An X In One Box Only)

CONTRACT

□ 110 Insurance
□ 120 Marine
□ 130 Miller Act
□ 140 Negotiable Instrument
□ 150 Recovery of Overpayment & Enforcement of Judgment
□ 151 Medicare Act
□ 152 Recovery of Defaulted (Excl. Veterans)
□ 153 Recovery of Overpayment of Veteran's Benefits
□ 160 Stockholders' Suits
□ 190 Other Contract
□ 195 Contract Product Liability

REAL PROPERTY

□ 210 Land Condemnation
□ 220 Foreclosure
□ 230 Rent Lease & Ejectment
□ 240 Torts to Land
□ 245 Tort Product Liability
□ 290 All Other Real Property

TORTS

PERSONAL INJURY

□ 310 Airplane
□ 315 Airplane Product Liability
□ 320 Assault, Libel & Slander
□ 330 Federal Employers Liability
□ 340 Marine
□ 345 Marine Product Liability
□ 350 Motor Vehicle
□ 355 Motor Vehicle Product Liability
□ 360 Other Personal Injury

PERSONAL INJURY

□ 362 Personal Injury - Med Malpractice
□ 365 Personal Injury - Product Liability
□ 368 Asbestos Personal Injury Product Liability

PERSONAL ROPERTY

□ 370 Other Fraud
□ 371 Truth in Lending
□ 380 Other Personal Property Damage
□ 385 Property Damage Product Liability

CIVIL RIGHTS

□ 441 Voting
□ 442 Employment
□ 443 Housing/ Accommodations
□ 444 Welfare
□ 440 Other Civil Rights

PRISONER PETITIONS

□ 510 Motions to Vacate Sentence
Habeas Corpus
□ 530 General
□ 535 Death Penalty
□ 540 Mandamus & Other
□ 550 Civil Rights
□ 555 Prison Condition

FORFEITURE/PENALTY

□ 610 Agriculture
□ 620 Other Food & Drug
□ 625 Drug Related Seizure of Property 21 U.S.C. 881
□ 630 Liquor Laws
□ 640 R R & Truck
□ 650 Airline Regs
□ 660 Occupational Safety/Health
□ 690 Other

LABOR

□ 710 Fair Labor Standards Act
□ 720 Labor/Mgmt Relations
□ 730 Labor/Mgmt. Reporting & Disclosure Act
□ 740 Railway Labor Act
□ 790 Other Labor Litigation
□ 791 Empl Ret Inc Security Act

BANKRUPTCY

□ 422 Appeal 28 U.S.C. 158
□ 423 Withdrawal 28 U.S.C. 157

PROPERTY RIGHTS

□ 820 Copyrights
☒ 830 Patent
□ 840 Trademark

SOCIAL SECURITY

□ 861 HIA (1395ff)
□ 862 Black Lung (923)
□ 863 DIWC/DIWW (405(g))
□ 864 SSID Title XVI
□ 865 RSI (405(g))

FEDERAL TAX SUITS

□ 870 Taxes (U.S. Plaintiff or Defendant)
□ 871 IRS - Third Party 26 U.S.C. 7609

OTHER STATUTES

□ 400 State Reapportionment
□ 410 Antitrust
□ 430 Banks and Banking
□ 450 Commerce/ICC Rates, etc.
□ 460 Deportation
□ 470 Racketeer Influenced and Corrupt Organizations
□ 810 Selective Service
□ 850 Securities/Commodities/ Exchange
□ 875 Customer Challenge 12 U.S.C. 3410
□ 891 Agricultural Acts
□ 892 Economic Stabilization Act
□ 893 Environmental Matters
□ 894 Energy Allocation Act
□ 895 Freedom of Information Act
□ 900 Appeal of Fee Determination Under Equal Access to Justice
□ 950 Constitutionality of State Statutes
□ 890 Other Statutory Actions

IV. ORIGIN (PLACE AN "X" IN ONE BOX ONLY)

☒ 1 Original Proceeding
□ 2 Removed from Court
□ 3 Remanded from Appellate Court
□ 4 Reinstated or Reopened
□ 5 Transferred from another district (specify)
□ 6 Multidistrict Litigation
□ 7 Appeal to District Judge from Magistrate Judgment

VI. CAUSE OF ACTION

(CITE THE U.S. CIVIL STATUTE UNDER WHICH YOU ARE FILING AND WRITE BRIEF STATEMENT OF CAUSE DO NOT CITE JURISDICTIONAL STATUTES UNLESS DIVERSITY.):

Brief description of cause:
35 U.S.C. §§ 101 *et seq.*, declaratory judgment action for patent noninfringement

VII. REQUESTED IN COMPLAINT:

CHECK IF THIS IS A CLASS ACTION UNDER F.R.C.P. 23 □ YES ☒ NO

DEMAND $

Check YES only if demanded in complaint
JURY DEMAND: ☒YES □ NO

VIII. RELATED CASE(S) (See instructions) **IF ANY**

JUDGE:

DOCKET NUMBER:

DATE 2/16/2008

SIGNATURE OF ATTORNEY OF RECORD

FOR OFFICE USE ONLY

RECEIPT # ____________ AMOUNT ____________ APPLYING IFP ____________ JUDGE ____________ MAG. JUDGE ____________

INSTRUCTIONS FOR ATTORNEYS COMPLETING CIVIL COVER SHEET FORM JS-44

Authority For Civil Cover Sheet

The JS-44 civil cover sheet and the information contained herein neither replaces nor supplements the filings and service of pleading or other papers as required by law, except as provided by local rules of court. This form, approved by the Judicial Conference of the United States in September 1974, is required for the use of the Clerk of Court for the purpose of initiating the civil docket sheet. Consequently a civil cover sheet is submitted to the Clerk of Court for each civil complaint filed. The attorney filing a case should complete the form as follows:

I. (a) **Plaintiffs - Defendants**. Enter names (last, first, middle initial) of plaintiff and defendant. If the plaintiff or defendant is a government agency, use only the full name or standard abbreviations. If the plaintiff or defendant is an official within a government agency, identify first the agency and then the official, giving both name and title.

(b) **County of Residence**. For each civil case filed, except U.S. plaintiff cases, enter the name of the county where the first listed plaintiff resides at the time of filing. In U.S. plaintiff cases, enter the name of the county in which the first listed defendant resides at the time of filing. (NOTE: In land condemnation cases, the county of residence of the "defendant" is the location of the tract of land involved).

(c) **Attorneys**. Enter firm name, address, telephone number, and attorney of record. If there are several attorneys, list them on an attachment, noting in this section "(see attachment)."

II. **Jurisdiction**. The basis of jurisdiction is set forth under Rule 8(a), F.R.C.P., which requires that jurisdictions be shown in pleadings. Place an "X" in one of the boxes. If there is more than one basis of jurisdiction, precedence is given in the order shown below.

United States plaintiff. (1) Jurisdiction is based on 28 U.S.C. 1345 and 1348. Suits by agencies and officers of the United States are included here.

United States defendant. (2) When the plaintiff is suing the United States, its officers or agencies, place an "X" in this box.

Federal question. (3) This refers to suits under 28 U.S.C. 1331, where jurisdiction arises under the Constitution of the United States, an amendment to the Constitution, an act of Congress or a treaty of the United States. In cases where the U.S. is a party, the U.S. plaintiff or defendant code takes precedence, and box 1 or 2 should be marked.

Diversity of citizenship. (4) This refers to suits under 28 U.S.C. 1332, where parties are citizens of different states. When Box 4 is checked, the citizenship of the different parties must be checked. (See Section III below; federal question actions take precedence over diversity cases.)

III. **Residence (citizenship) of Principal Parties**. This section of the JS-44 is to be completed if diversity of citizenship was indicated above. Mark this section for each principal party.

IV. **Cause of Action**. Report the civil statute directly related to the cause of action and give a brief description of the cause.

V. **Nature of Suit**. Place an "X" in the appropriate box. If the nature of suit cannot be determined, be sure the cause of action, in Section IV above, is sufficient to enable the deputy clerk or the statistical clerks in the Administrative Office to determine the nature of suit. If the cause fits more than one nature of suit, select the most definitive.

VI. **Origin**. Place an "X" in one of the seven boxes.

Original Proceedings. (1) Cases which originate in the United States district courts.

Removed from State Court. (2) Proceedings initiated in state courts may be removed to the district courts under Title 28 U.S.C. Section 1441. When the petition for removal is granted, check this box.

Remanded from Appellate Court. (3) Check this box for cases remanded to the district court for further action. Use the date of remand as the filing date.

Reinstated or Reopened. (4) Check this box for cases reinstated or reopened in the district court. Use the reopening date as the filing date.

Transferred from Another District. (5) For cases transferred under Title 28 U.S.C. Section 1404(a). Do not use this for within district transfers or multidistrict litigation transfers.

Multidistrict Litigation. (6) Check this box when a multidistrict case is transferred into the district under authority of title 28 U.S.C. Section 1407. When this box is checked, do not check (5) above.

Appeal to District Judge from Magistrate Judgment. (7) Check this box for an appeal from a magistrate's decision.

VII. **Requested in Complaint**. Class Action. Place an "X" in this box if you are filing a class action under Rule 23, F.R.Cv.P.

Demand. In this space enter the dollar amount (in thousands of dollars) being demanded or indicate other demand such as a preliminary injunction.

Jury Demand. Check the appropriate box to indicate whether or not a jury is being demanded.

VIII. **Related Cases**. This section of the JS-44 is used to reference relating pending cases if any. If there are related pending cases, insert the docket numbers and the corresponding judge names for such cases.

Date and Attorney Signature. Date and sign the civil cover sheet.